Universidad Carlos III de Madrid

PhD Dissertation

New Approaches to Interactive Multimedia Content Retrieval from different Sources

Author: Julián Moreno Schneider

Advisor: Paloma Martínez

PhD Program in Computer Science and Technology

Computer Science Department

Leganés, October 2015

DOCTORAL THESIS

New Approaches to Interactive Multimedia
Content Retrieval from different Sources

Author: JULIÁN MORENO SCHNEIDER
Director: PALOMA MARTÍNEZ

Signature from PhD committee

Signature

1. President: Ana García Serrano

2. Secretary: Belén Ruiz Mezcua

3. Vocal: Davide Buscaldi

Grade:

Leganés, 19 October 2015

To all the people that can describe me as the orderly disorder.

Acknowledgements

I would like to acknowledge my wife in first place for helping and supporting me all the time I have spend with this PhD. She has been supporting me in good and bad times. Its availability has been constant 24/7. For all those afternoons in the park, all these movies at the cinema, for all those pointless conversations ... for all: thank you very much.

Then my family has also given me all the help I needed in order to fulfill the tasks. Thank you very much.

All my colleges and bosses and specially to my advisor Paloma Martínez. I do not forget Isabel Segura and Lourdes Moreno by their advices and help at every moment and the people I knew in Colmenarejo, that made easier my time there. To Thierry for welcoming me as PhD internship‘and bring me his wisdom and friendship.

To Pablo and Ricardo for their unconditional friendship and for bringing me fresh air at the right time. To every person that had bet all his possessions to me. Grego, Chema, Homer ... for those moments that managed to distract me, the beers on the lawn before classes and such a good time in the spring festival. As a wise man said: 'El bebé ya está en la cuna'.

Also to my friends and all the people that I forget now but that also have been there.

Thank you very much.

Resumen

Los Sistemas Interactivos de Recuperación de Información Multimodal (IMIR) incrementan las capacidades de los sistemas tradicionales de búsqueda con la posibilidad de recuperar información de diferentes tipos (modos) y a partir de diferentes fuentes. El incremento del contenido en internet a la vez que la diversificación de los medios de acceso a la información (móviles, tabletas, relojes inteligentes) fomenta la necesidad cada vez mayor de este tipo de sistemas.

En esta tesis se ha definido un modelo formal para la descripción de sistemas de recuperación de información multimodal e interactivos que consultan varios motores de recuperación. Este modelo incluye la definición formal y generalizada de cada componente de un sistema IMIR, a saber: información multimodal organizada en colecciones, consulta multimodal, diferentes motores de recuperación, sistema de gestión de fuentes (handler), módulo de gestión de resultados (fusión) y las interacciones de los usuarios.

Este modelo se ha validado en dos escenarios. El primero, en un caso de uso focalizado en recuperación de información relativa a deportes. Se ha desarrollado un prototipo que implementa un subconjunto de todas las características del modelo: una colección multimodal que se relaciona semánticamente, tres tipos de consultas multimodal (texto, audio y texto + imagen), seis motores diferentes de recuperación (búsqueda de respuestas, búsqueda de texto completo, búsqueda basada en ontologías, OCR en imagen, detección de objetos en imagen y transcripción de audio), una estrategia de selección de fuentes basada en reglas definidas por expertos, una estrategia de combinación de resultados y el registro de las interacciones.

Se utiliza la medida NDCG (normalized discounted cumulative gain) para describir los resultados obtenidos por cada motor de recuperación. Estos

resultados son: $10,1\%$ (*Question Answering*), 80% (*Búsqueda a texto completo*) y $26,8\%$ (*Búsqueda en ontologías*). Estos resultados están en el orden de los trabajos del estado de arte considerando foros como CLEF (Cross-Language Evaluation Forum). Cuando se utiliza la combinación de motores de recuperación, el rendimiento de recuperación de información se incrementa en un porcentaje de ganancia de $771,4\%$ con *Question Answering*, $7,2\%$ con *Búsqueda a texto completo* y $145,5\%$ con *Búsqueda en ontologías*.

El segundo escenario es un prototipo centrado en recuperación de información de medios sociales en el dominio de salud. Se ha desarrollado un prototipo basado en el modelo propuesto y que integra información del dominio de salud generada por el usuario en medios sociales, bases de conocimiento, consulta, motores de recuperación, módulo de selección de fuentes, módulo de combinación de resultados y la interfaz gráfica de usuario. Además, los documentos incluidos en el sistema de recuperación han sido previamente anotados mediante un proceso de extracción de información semántica del dominio de salud.

Además, se han definido técnicas de adaptación de la funcionalidad de recuperación de un sistema IMIR analizando interacciones pasadas mediante árboles de decisión, redes neuronales y agrupaciones.

Una vez modificada la estrategia de selección de fuentes (handler), se ha evaluado de nuevo el sistema usando técnicas de clasificación. Las mismas consultas y juicios de relevancia realizadas por los usuarios en el primer prototipo sobre deportes se han utilizado para esta evaluación.

La evaluación compara la medida NDCG (normalized discounted cumulative gain) obtenida con dos enfoques diferentes: el sistema multimodal usando reglas predefinidas y el mismo sistema multimodal una vez que la funcionalidad se ha adaptado por las interacciones de usuario. El NDCG ha mostrado una mejoría entre $-2,92\%$ y $2,81\%$ en función de los métodos utilizados. Hemos considerado tres características para clasificar los enfoques: (i) el algoritmo de clasificación; (ii) las características de la consulta; y (iii) las puntuaciones para el cálculo del órden de los motores de recuperación. El mejor resultado se obtiene utilizando el algoritmo de clasificación basado

en probabilidades, las puntuaciones para los motores de recuperación basados en la media de la posición del primer resultado relevante y el modo, el tipo, la longitud y las entidades de la consulta . Su valor de NDCG es $81,54\%$.

Abstract

Interactive Multimodal Information Retrieval systems (IMIR) increase the capabilities of traditional search systems with the ability to retrieve information in different types (modes) and from different sources. The increase in online content while diversifying means of access to information (phones, tablets, smart watches) encourages the growing need for this type of system.

In this thesis a formal model for describing interactive multimodal information retrieval systems querying various information retrieval engines has been defined. This model includes formal and widespread definition of each component of an IMIR system, namely: multimodal information organized in collections, multimodal query, different retrieval engines, a source management system (handler), a results management module (fusion) and user interactions.

This model has been validated in two stages. The first, in a use case focused on information retrieval on sports. A prototype that implements a subset of the features of the model has been developed: a multimodal collection that is semantically related, three types of multimodal queries (text, audio and text + image), six different retrieval engines (question answering, full-text search, search based on ontologies, OCR in image, object detection in image and audio transcription), a strategy for source selection based on rules defined by experts, a strategy of combining results and recording of user interactions.

NDCG (normalized discounted cumulative gain) has been used for comparing the results obtained for each retrieval engine. These results are: $10,1\%$ (*Question answering*), 80% (*full text search*) and $26,8\%$ (*ontology search*). These results are on the order of works of the state of art considering forums like CLEF. When the retrieval engine combination is used, the information

retrieval performance increases by a percentage gain of $771,4\%$ with *question answering*, $7,2\%$ with *full text search* and $145,5\%$ with *Ontology search*.

The second scenario is focused on a prototype retrieving information from social media in the health domain. A prototype has been developed which is based on the proposed model and integrates health domain social media user-generated information, knowledge bases, query, retrieval engines, sources selection module, results' combination module and GUI. In addition, the documents included in the retrieval system have been previously processed by a process that extracts semantic information in health domain.

In addition, several adaptation techniques applied to the retrieval functionality of an IMIR system have been defined by analyzing past interactions using decision trees, neural networks and clusters.

After modifying the sources selection strategy (handler), the system has been reevaluated using classification techniques. The same queries and relevance judgments done by users in the sports domain prototype will be used for this evaluation.

This evaluation compares the normalized discounted cumulative gain (NDCG) measure obtained with two different approaches: the multimodal system using predefined rules and the same multimodal system once the functionality is adapted by past user interactions. The NDCG has shown an improvement between $-2,92\%$ and $2,81\%$ depending on the approaches used. We have considered three features to classify the approaches: (i) the classification algorithm; (ii) the query features; and (iii) the scores for computing the orders of retrieval engines. The best result is obtained using probabilities-based classification algorithm, the retrieval engines ranking generated with Averaged-Position score and the mode, type, length and entities of the query. Its NDCG value is **81,54%**.

Contents

List of Figures

LIST OF FIGURES

LIST OF FIGURES

List of Tables

Glossary

AENN All that an Entity Needs is a Name

AENN Auto-Encoder Neural Network

AESS Adaptive Exploratory Search System

AI Artificial Intelligence

AMR Adaptive Multimedia Retrieval

AP Average Precision

API Application Programming Interface

ARS Averaged Ranking Score

AT Audio Transcription

BIMIR Biomedical Interactive Multimodal Information Retrieval

CG Conceptual Graphs

CLEF Cross Language Evaluation Forum

DAG Directed Acyclic Graph

DL Descriptions Logic

DL Digital Libraries

DNS Dragon Naturally Speaking

ESS Exploratory Search System

FT Full Text search result

FTS Full Text search

FUS First Used Score

GUI Graphical User Interface

HCI Human-Computer Interaction

iCLEF Interactive track of Cross-Language Evaluation Forum

IIR Interactive Information Retrieval

IMIR Interactive Multimodal Information Retrieval

IR Information Retrieval

IRF Inverse Resource Frequency

IS Interactions-based Score

ISB Information Seeking Behavior

ISR Information Systems Resources

ISR Inverse Square Ranking

JCD Joint Composite Description

LRS Lowest Ranking Score

MAP Mean Average Precision

MathS Mathematical Score

MBIR Metadata-Based Information Retrieval

MIR Multimodal Information Retrieval

MRR Mean Reciprocal Rank

NDCG Normalized Discounted Cumulative Gain

ObS Ontology-based search

OCR Optical Character Recognition

OCRI Optical Character Recognition in Image

ODI Object Detection in Image

ODP Open Directory Project

OLD Open Linked Data

ONT Ontology-based search result

OOR Object-Oriented Representation

PoI Points of Interest

QA Question Answering result

QAS Question Answering search

R-P R Precision

R&D Research and Development

GLOSSARY

RDF Resource Description Framework

RDFS Resource Description Framework Schema

RDQL RDF query language

RE Retrieval Engine

RET Recognition Evaluation Tool

RFF Rank-based Fusion Function

RRF Regular Random Forests

RUCoD Rich Unified Content Description

SCD Spatial Color Distribution

SE Search Engine

SIRE Social Image Retrieval Engine

SMO Sequential Minimal Optimization

SW Semantic Web

TF-IDF Term Frequency - Inverse Document Frequency

TREC Text Retrieval Evaluation Conference

TWF Term Weight Factor

TWF Term Weight Frequency

UCAIR User-Centered Adaptive Information Retrieval

UM User Modelling

WSR Windows Speech Recognition

1

Introduction

Internet is getting huge and there is too much information that must be managed (stored and requested). Most of this information is multimedia. This is due to the new available devices, such as smartphones or tablets, which allow online multimedia information generation by simply pressing a button. Users accessing this amount of information is a real problem, because many different search systems must be requested separately. In order to avoid the need of requesting several systems, a proposal that manages to combine several search systems (transparently to the user) offering a single results' set merging outcomes from the different multimedia sources is described in this thesis.

1.1 Motivation

Current society is characterized by a constant technological revolution, where the generation and consumption of information is reaching huge levels. Internet, the main information container, is increasing its content exponentially. Since 1991 (the date of the creation of world wide web when a single web page existed) internet has expanded the amount of content available to take in 2013 approximately 975 million web pages[1].

Besides the available information, internet traffic has grown to the same extent. Cisco anticipates that '*Global IP traffic will reach 1.1zettabytes per year or 91.3 exabytes*

[1]Data extracted from http://www.internetlivestats.com/total-number-of-websites/ at 23/07/2015

(one billion gigabytes) per month in 2016. By 2018, global IP traffic will reach 1.6 zettabytes per year, or 131.6 exabytes per month'[2].

Using the *one second monitor*[3], it is noted that 28.714 GB of traffic is currently generated in one second (measured at 22/07/2015 16:39). Besides the amount of information, some other significant data generated by one second are: 2.412.483 emails sent, 104.611 YouTube videos viewed, 49.801 Google searches, 1.814 Skype calls, 2.084 Tumblr posts, 2.387 Instagram photos uploaded and 9.673 Tweets sent.

Focusing on the data format preferred by users, services such as Google[4] (specialized in text content), YouTube[5] to search for videos, Flicker[6] to publish and search for photos or SoundCloud[7], a social network for music sharing, among others, are well known. When dealing with audio, image or video, these commercial systems are mainly based on the characterization of resources using textual metadata, which are later matched against user query expressions. Some examples of metadata of documents are the *'author'*, *'date of creation'*, *'title'*, *'language'* and others. Furthermore, there is a clear need to apply semantic web tools and resources to improve the retrieval results as far as documents in different formats are concerned and certain knowledge about the objects (such as meaning, purpose, etc) is required. This improvement can be achieved by using semantic relations among documents (instantiated by means of named entities appearing inside elements and relationships among these entities). In this way, Google has launched Knowledge Graph[8] in order to show data and documents semantically related to the terms used in the query. Another way to exploit semantic search is the approach followed by Facebook, Graph Search[9], where a Facebook user can also express, explicitly and implicitly, some restrictions such as the preferred geographic location or some specific interests when searching for people.

Thus, retrieval methods do not remain constant and become dependent on: the device used to query (PC, smartphone, tablet, etc.), what is being queried and who is

[2]http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html accessed at 23/07/2015

[3]http://www.internetlivestats.com/one-second/ accessed at 23/07/2015

[4]https://www.google.com

[5]https://www.youtube.com/

[6]https://www.flickr.com/

[7]https://soundcloud.com/

[8]https://www.google.com/intl/es/insidesearch/features/search/knowledge.html accessed at 23/07/2015

[9]https://www.facebook.com/about/graphsearch accessed at 23/07/2015

querying. Besides, advances in devices available to users are leading to a change in the formats applied in the definition of queries. Google has introduced voice query (users can interact with the search engine by using a microphone to formulate the query) and, previously, they included queries through images (searching images that are similar to the image of the query).

Figure 1.1 shows the evolution that has followed internet since its inception. As can be seen, at first there were only a couple of protocols. The next major advance is the arrival of desktop PC while then the first online services began to develop (around 1989). The web is then created (in 1991), denominated Web1.0 and being the forerunner of what we now know as the Internet. Web1.0 consisted of web pages and, mostly static, technologies (*HTTP, HTML*) and standards.

The development of programming languages such as HTML, java or flash, evolved the internet to its second version (Web2.0) also called *'social web'*. This name was assigned by the emergence of social content such as blogs or wikis where users had a more active role generating web contents. Some examples are wiki pages, such as Wikipedia[10] where users can publish their own content (although it is later revised) or personal blogs, which are used as public diaries.

Technological advances were the reason for a new evolution in internet, leading to the *'real time web'*. The next version of the Web (Web3.0) is characterized by the appearance of semantic knowledge (*'semantic web'*). The semantic content has burst onto the web by the popularization of knowledge management systems as ontologies or taxonomies. In that sense, a project currently stands, Linking Open Data[11], which is responsible for grouping semantic knowledge in internet by unifying a lot of ontologies (570 data sets are connected by 2909 links' sets).

The next evolution of the internet (Web4.0 or future web) seems to be directed toward what is called *'intelligent web'*. This advance symbolizes a web in which the systems will be able to think for themselves and adapt to the needs and desires of the users.

This is not only because there is much more information available, but there are many more internet users. The number of users connected to internet in 2014 world-wide approaches 2.095 billion people, whereas in 1995 it was 16 million and 147 million

[10] https://www.wikipedia.org/

[11] http://linkeddata.org/ accessed at 23/07/2015

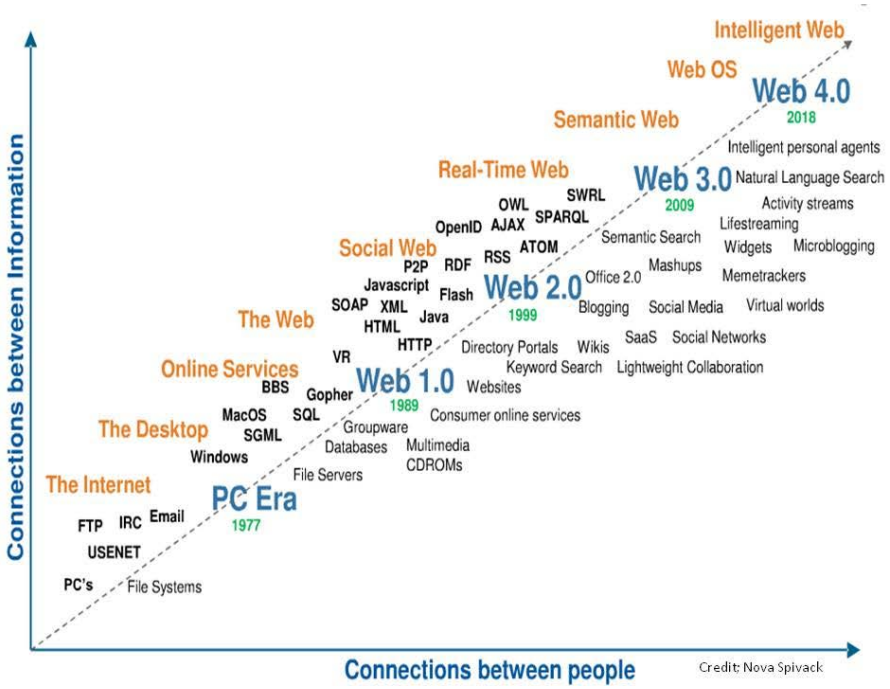


Figure 1.1: Internet evolution through time from Web1.0 to the future web (Web4.0) (source: http://www.slideshare.net/novaspivack/web-evolution-nova-spivack-twine.html)

in 1998[12]. The geographical distribution is (divided into continents)[13]: 272M (78.9% population) in North America, 216M (37.74%) in Latin America, 476M (64.5%) in Europe, 922M (22.14%) in Asia, 119M (11.6%) in Africa and 21M (58.19%) in Oceania. Although still 60% of the world population has no internet connection[14] (in Asia and Africa the penetration is low), the absolute number of user is very large in every continent.

In addition to the growing amount of information and users, devices are very diverse and move away from traditional PCs. According to some studies by consultancy firms [15],[16] PC shipments have fallen 9.5 percent in 2013 and 2.9 percent in 2014, while sales

[12]Numbers extracted from http://www.allaboutmarketresearch.com/internet.htm at 23/07/2015

[13]Numbers extracted from http://www.go-gulf.com/blog/online-time/ at 23/07/2015

[14]http://www.latimes.com/business/technology/la-fi-tn-60-world-population-3-billion-internet-2014-20140507-story.html accessed at 23/07/2015

[15]http://techcrunch.com/2014/07/06/gartner-device-shipments-break-2-4b-units-in-2014-tablets-to-overtake-pc-sales-in-2015/ accessed at 23/07/2015

[16]http://www.extremetech.com/computing/185937-in-2015-tablet-sales-will-finally-surpass-pcs-fulfilling-steve-jobs-post-pc-prophecy accessed at 23/07/2015

of smartphones and tablets have increased 3, 9 and 23.9 percent respectively in 2014.

Smartphones (80% of internet users have one[17]) are not the only device that has appeared as a novelty for internet access, but there are others that are becoming popular as smartwatches (9%), smart tvs (34%), games consoles (37%), smart wristbands (7%), although tablets (47%) rank as the best candidate. The tablets are portable devices with large screens that allow a connection (WiFi or GSM) to the internet through which a user can perform virtually the same operations as (s)he can do in a PC. This, coupled with its low cost, have made it a very attractive device and whose sales will account (along with smartphones) 87% of connectable devices in 2017[18].

The use of smartphones has exceeded (first time ever) the use of traditional computers to check internet. This is because many people have smartphones (with data connection) but neither computer nor internet access at home. In addition to the fact that smartphones have allowed internet access to many more people than traditional PCs, internet usage on smartphones is different from traditional computer usage. Mobile devices usage and their capabilities as multimedia devices (users can take pictures or record high definition video), along with internet connection, have also helped to the increase of online multimedia content.

Users should find all the information they need easily and without having to request several sources. In order to achieve this, the definition and creation of systems that allow the request of information from different sources arise. In addition, users also want to search in a more complex way with different modes, so new features have to be contemplated. How can users pose queries that combine different media (voice, text, image, etc.)? How can users select several sources to search for information? How should users receive the results considering that they come from different sources and have different formats?

All these questions have the same answer: *users need to make multimodal requests to multimodal search engines which should return the best information, from the most suitable source(s) and in the correct format from all the available information elements.*

[17]http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/ accessed at 23/07/2015

[18]Data extracted from http://www.forbes.com/sites/louiscolumbus/2013/09/12/idc-87-of-connected-devices-by-2017-will-be-tablets-and-smartphones/ accessed at 23/07/2014

1.2 Problem description

The main problem addressed in this thesis is motivated by the growing presence of multimedia content on the Internet; users need to access bigger and bigger amounts of information in different formats (such as video, text, audio, images, graphics, etc.) and sources in a faster and easier way.

The huge amount of available information is handled by different information retrieval systems depending on the format (text, audio, video, image or combinations) of the information or the type of response needed (a specific data, a document, an image, etc.). Of course, the user cannot distinguish among all these systems and, obviously, (s)he is not intended to do it. For this reason, a big challenge is to provide a user-transparent information retrieval system. This transparency is defined as: *the user does not care about the retrieval method behind the system nor the type of answer returned nor the way these results are combined to build a unique answer.*

Furthermore, there is a great variety of users using search systems with diverse needs. Considering that each user wants to obtain different information, both in type and format, the problem takes a dimension even higher. It is not only important to adapt the results to the user, but also to allow requesting easily complex systems to access huge volumes of information requesting different kinds of systems (or modes) and offering the possibility of domain-specific or general domain systems.

It is possible to envisage different scenarios where this multimodal retrieval technology solves user needs. Suppose a user wants to ask for a job in a company called 'Blanco' (White), but he knows nothing more about the company, so he will use its name as a query. The system will return relevant information about the company, but also some 'useless' information talking about 'the color' and the ex-former Spanish Minister 'José Blanco'. The user is not only interested in the company's official website, but also in blogs where people talk about the working conditions. Even a map of the location of the offices to attend in person could be interesting.

Music domain also offers illustrative examples. A user has a song (a file) and (s)he knows the artist. In this case the user wants to know which record the song belongs to, in order to acquire it. The user would make a query composed of two elements: text and audio file such as record from 'author' where the song (file) appears. The user expects to get the name of the record, although it could be useful to provide also the

cover, other songs from the same record or even a website where to buy this song or the record (Fnac[19] or iTunes[20])

Another interesting scenario is to help the automatic production and generation of audiovisual content. If we consider a scenario where a journalist (sports editor at a television station) has to prepare news about a F1 race, (s)he has to cover information from all F1 races, traveling to all F1 Grand Prix for live broadcasts. Around each of these trips, (s)he has to document and archive all audiovisual material captured in a race. At the same time (s)he must develop additional pieces of information related to the last race and audiovisual productions to publish them into news broadcasts. Retrieving these pieces of information is a hard task that can be simplified by a multimodal retrieval system. In a more familiar scenario, a parent wants to catch up on current events and sports, while his two sons want to get educational television programs. Using a multimodal retrieval system they can visualize information in just a few seconds on its IP TV terminal.

Another interesting scenario is e-learning which makes use of multimedia information to provide knowledge that is becoming increasingly complex. Teachers offer students some multimedia documents of any kind, but students could need further (multimodal) information which could be easily found using a multimodal retrieval system.

An example of a commercial application is intellectual property management or marketing information tracking by means of monitoring audiovisual content. Its goal is to audit the use of audiovisual content in Web pages, radio, television and even in clubs and discotheques. Multimedia content is charged by royalties, and whenever a multimedia content is used, some payment must be done to its authors. Thus, it is interesting for authors to monitor if they are given the appropriate amount of money for their multimedia content. A multimodal retrieval system could provide detailed information about: the performers, the channels that have been issued and in what time intervals; besides offering charts, lists of videos or more detailed information about singers could provide an interesting scenario for monitoring companies.

The problem we want to address with this thesis is that the retrieval of multimodal information currently has to be done from different sources. This retrieval must be

[19] http://www.fnac.es/home/music.aspx
[20] https://www.apple.com/es/itunes/

simple, quick and transparent to the user, something that now is not been doing. Web search engines are the most similar existing systems, but they request every 'available'[21] and integrate different sources in a simple results' list. We will define a multimodal information retrieval model that requests multiple heterogeneous sources making emphasis in two aspects:

- What sources are requested. Depending on the query, it may require requesting different systems and the user should not know that. Decide which sources are requested in each case is the first part of the problem we want to address.
- The combination of results is the second part of the problem. By requesting different sources, results of each one are obtained and they can be heterogeneous. The results should be managed to provide the user with information in the most intuitive way. Which is the best combination approach is the second problem we address in this thesis.

1.3 Fixing the terminology

A set of definitions is given in this section with the purpose of clarifying terminology used in this thesis in order to achieve a better understanding of the proposal.

First of all, note that an information retrieval (IR) system is composed of a query, indexing and matching processes, documents and retrieved objects. The generic architecture of an IR system is shown in figure 1.2. The works of the state of art name differently the components of an IR system. Then we will try to fix the terminology to make the reading of the thesis more affordable.

The information retrieved by this type of systems is called **multimodal information**. The concept of multimodality has two meanings. First, it is used when the information is present in various formats, also known as modes (for instance, text documents, images, etc). Second, it is also applied when interactions in different modes take place such as touch and voice.

From now on, we will use multimodality referring to having various formats (modes). These modes range from text to multimedia elements as audio, video, images, 3D

[21] Available means every retrieval engine that accepts this mode of query, i.e. if the query is a text, every retrieval engine that accepts a text query will be requested.

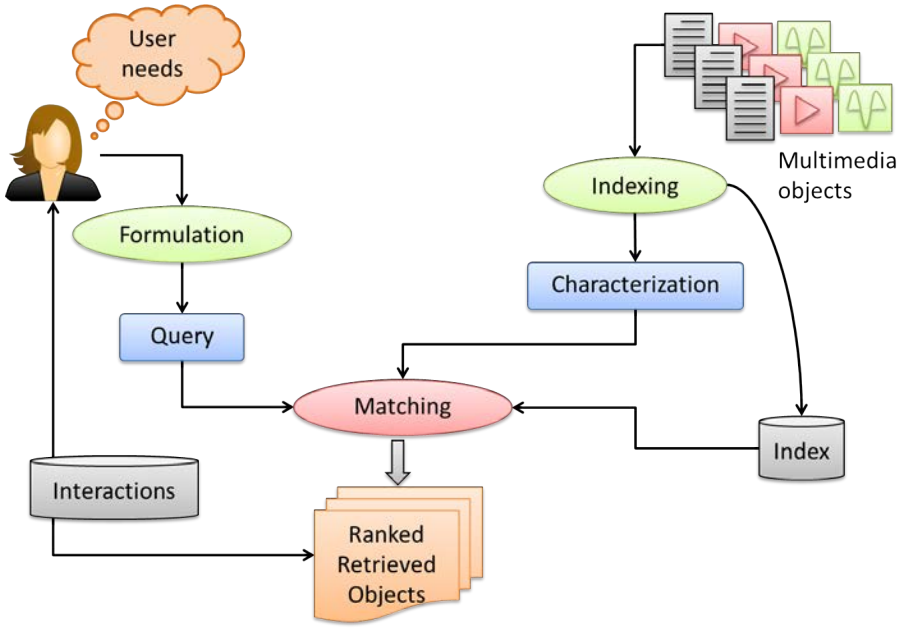


Figure 1.2: Schema of the generic information retrieval process

sketches, etc. An information collection is defined as a set of documents which normally have common characteristics or properties such as the *topic* the document is about, the *domain* the documents belongs to (sports, biomedicine, financial, etc), the *length* (or *size*) of the documents, the structure the documents have (XML, plan text, MPEG-7 video), etc. In this sense, a collection that contains documents in different modes will be known as **Multimodal Collection** (Arampatzis et al. [2011], Yilmaz et al. [2012] and Yang et al. [2002]). The same type of collection is named differently in other research works: *Multimedia Collection* (Kludas et al. [2008] and Yang et al. [2002]) or *Mixed Collection* (Galiano et al. [2007]).

A query is defined as a set of multimedia elements. If a query contains elements of one mode, it is known as query, simple query or **monomodal query**. On the contrary, if it contains elements of more than one mode, it is a **multimodal query**. This concept has also received other denominations. Some studies called it *combined query* [Nottelmann and Fuhr, 2003; Yang et al., 2002] or *hybrid query* [Demner-Fushman et al., 2012] when it concerns only two modes such as text or image, or as *multimedia query* because its components can be multimedia elements as in de Vries [1998].

An information retrieval system is characterized by its functionality: it retrieves

information from sources (any number or type) based on an input query. In this work, as well as in most of the related literature, every retrieval system is called **Retrieval Engine** (RE). This is due to the fact that its functionality is to execute a process of information retrieval by making a matching between the query and the information available. Other works refer to them as *search engines*, but mostly when talking about Web environment. They are also known as *retrieval servers* in federated search[22] [Nottelmann and Fuhr, 2003] and *retrieval systems* [Miguel and Magalhes, 2008] or *search services or verticals*[23] in the field of Aggregated Search[24] [Arguello et al., 2011].

A Multimedia Information Retrieval (MIR) system is also characterized by the ability of using different information retrieval engines (REs) simultaneously. As far as information retrieval from several retrieval engines is concerned, the matching process (in figure 1.3) contains two specific components (handler and results' fusion) that help the management of several retrieval engines. In this thesis the element that decides which REs are requested by each query is known as **Handler**. Nevertheless, in the literature it is referred to by different names: *server selection module* in federated search [Hong and Si, 2012], *orchestrator* in Paris et al. [2010], *mediator or dispatcher* in Nottelmann and Fuhr [2003], or *broker* in Chernov et al. [2006].

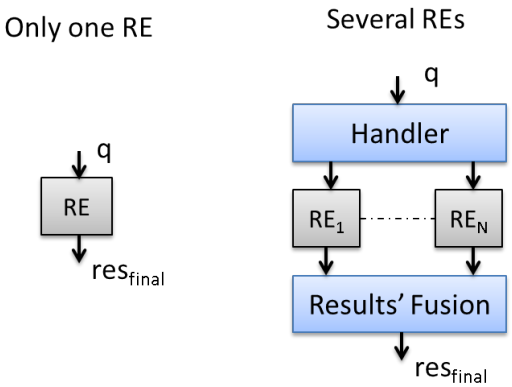


Figure 1.3: Difference of matching process between single and multiple REs

[22]Federated search systems covers information retrieval from several retrieval engines. A complete description of this field and the differences with the approach of this thesis are presented in section 2.8.

[23]Vertical is more used for different retrieval engines inside a portal such as Yahoo! and its verticals: web, images, maps, blogs, etc. Each of these retrieval engines is known as a vertical.

[24]The main goal of *Aggregated search* is to offer users 'structures' of information from different sources or results to complete their information needs (or queries).

The results obtained by the different REs have to be organized before being presented to the user. This process is composed of three parts: *Combination* or *Aggregation*, that joins the results coming from different REs; *Selection* or *Filtering*, that applies some filters to discriminate not relevant results; and *Reranking*, that reorganizes the results' sets. In this work this module will be known as **Results' Fusion Module** or **Results' Fusion**.

The concept of *interactivity* has different interpretations in the revised literature. In most of the works interactivity means that users have an active role in implementing the system, and participating through choices or decisions they make. By contrast, we will consider interactivity as the process of exchange between a user and the system.

A *user* is any client (person, external system, etc.) using a RE. The system registers actions performed by the users. Each action that users make in the system is called an **Interaction**. These actions are considered interactions between the user and the system. Therefore, the interactivity of the system is directly related to the actions carried out by users.

1.4 Objectives

An interactive multimodal information retrieval system requires services and resources for the management of multimedia content and several retrieval engines. The fact that many commercial services and prototypes use only one engine is a solid evidence for the complexity of a system that can handle every multimedia mode without distinction. On the contrary, there are online search services, recently developed, that perform searches on many different engines. Some examples are travel search engines, such as *Expedia*[25], *Skyscanner*[26] or *Kayak*[27], that offer results from different travel providers. If a user makes a search, (s)he obtains results from different engines: flights, hotels, etc. The results are quite structured and they contain the same information. The idea is to extend the paradigm they are using to apply it for any domain.

When dealing with technology that request several retrieval engines there are two problems that arise with every considered engine: when is it requested and how are its results processed. Most techniques rely on the mode of the query to select engines and

[25]http://www.expedia.es/
[26]http://www.skyscanner.es/
[27]http://www.kayak.es/

a simple mixture as a final list as combination, and therefore, they do not really adapt to different environments or queries. Furthermore, new techniques may be developed if we would like to work with several multimodal engines.

Our approach attempts to model an interactive multimodal information retrieval (IMIR) system using the most relevant components and their characteristics. We assume that the engine selection and results combination criteria would cover different types of queries. But this is not enough, we need to be able to identify the behavior of users and the way this behavior can be exploited by an IMIR system. In contrast with other approaches, we attempt to use semi-supervised machine learning like decision trees and neural networks in order to reduce the need for specific manual definitions. In turn, our aim is to exploit the past interactions of an IMIR system through the use of classification algorithms in order to avoid the need for expert defined rules.

We summarize the objectives of this thesis like:

- **Describe a formal model for the definition of interactive multimodal information retrieval systems.** Describe the architecture of an IMIR system and conceptualize (define formally) every component that is present in an IMIR system. Having a model allows interoperability and scalability of independently developed components.

- **Design and develop a first Interactive Multimodal Information Retrieval system based on the formal model**. Information retrieval has been well studied, but mostly applied as monomodal systems. Regarding systems that request several retrieval engines, they are mainly studied in two research areas: federated search and aggregated search. We will attempt to transfer techniques from these works to interactive multimodal IR using semantically related collections. This prototype does not include every feature of the model, but a subset. Nevertheless, this prototype will be used as the validation of the formal model. This prototype will also serve as a starting point for user interaction registration.

- **Study adaptation of retrieval functionality to user behavior.** The purpose of the adaptation is to offer better results for an information need by means of using the user interactions information. An important part of this improvement will be the handler functionality, which will be modified based on user interactions,

so the number, order and type of requested sources will change. Furthermore, how results are displayed and in what order also must be adapted. In order to adapt the retrieval functionality with historical interactions we need a method to extract user behavior patterns from these interactions. The adaptation is to be designed using Artificial Intelligence (AI) machine learning techniques. The scores will use information about relevance of documents and rankings. As machine learning techniques, there are some that seem interesting such as decision trees or neural networks.

- **Implement and evaluate a system adapting its retrieval functionality to past interactions (user behavior).** The system adapts its functionality to user behavior applying the results of the study previously done. The system should be an extension of the basic prototype including functionality adaptation based on historical interactions. The best way to evaluate an IR system is by carrying out comparative evaluations through evaluation forums. In this sense, there are several evaluation forums that could be interesting (interactive Cross-Language Evaluation Forum - iCLEF, Interactive Track 2005 TREC (Text Retrieval Evaluation Conference) or Web Track 2013 TREC among others). The most suitable evaluation forum for our purpose is 'Federated Web Search Track', that is introduced in section 2.8. Multimodal information was enabled in Federated Web Search Track[28] by adding multimedia retrieval engines. This could serve as a comparative evaluation.

- **Develop a second Interactive Multimodal Information Retrieval system based on the formal model**. This second IMIR prototype works in a new domain to test the adaptability of the model to the characteristics of new domains. The main objective of this second prototype is to test the capabilities of the model to define systems systems in different specific domains, so it works on a new domain.

[28]https://sites.google.com/site/trecfedweb/ accessed at 23/07/2015

1.5 Proposed solution

The solution developed is based on an incremental methodology, where a first approach is subsequently adapted and improved to obtain an extended approach. The particularized methodology is shown in figure 1.4. This figure shows that the methodology is composed of six tasks: the definition of a formal model of interactive multimodal information retrieval, the implementation of two prototypes based on this model (one in the sports domain and one in the health domain), the evaluation of this first prototype, the adaptation made to the functionality of the prototype according to the registered interactions and the evaluation of the extended prototype (including adaptation techniques).

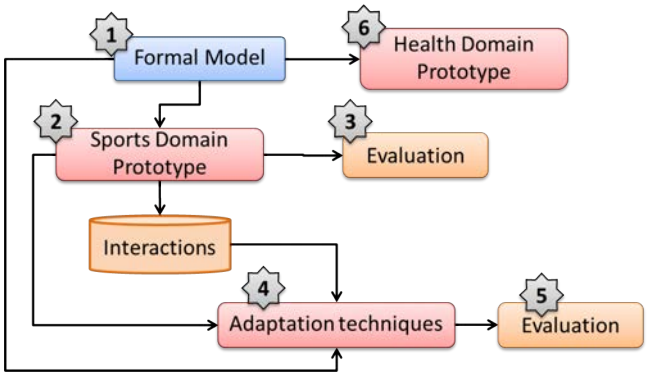


Figure 1.4: Particularized methodology implemented for the development of this thesis

- The first task of the thesis accomplishment is the **definition of a formal model for describing interactive multimodal information retrieval systems** that request several retrieval engines. This model should include every essential components of an IMIR system making a formal and generalized definition of them. The components of the model are: multimodal information organized in collections, query (multimodal), different retrieval engines, 'handler' and 'fusion' modules and user interactions. The whole model is described in chapter 3. It is important to define a model because defining a system based on a model provides a competitive advantage over other systems: each module which is defined according to the model can be automatically included or exchanged. This facil-

itates the scalability of the system, and allows anyone to define new modules or modifications to systems that are defined using the model.

- The second task is the **development of a prototype in the sports domain** based on the formal model. The prototype (described in chapter 4) implements only a subset of all the features of the model. The implemented features are: a multimodal collection that is semantically related, three types of multimodal queries (text, audio, and text + image), six different retrieval engines (question answering, full text search, ontology-based search, Optical Character Recognition (OCR) in image, object detection in image and audio transcription), a fusion strategy based on expert-defined rules, a round-robin-based results' combination strategy and the registration of interactions. The developments performed by this thesis are to be applied to an IMIR system which should be operational. In addition, the system should also have the functionality to record user interactions. These interactions will be later used (in the fourth task) to extract user behavior.

- The **evaluation of the sports-domain prototype** is the third task of this thesis. The prototype validation goal is twofold. On the one hand, it tries to validate the performance of the retrieval system using standard IR measurements (presented in chapter 5). On the other hand, every interaction performed by users is logged during the evaluation (see section 6.3) to use it for the adaptation mechanisms. The evaluation will allow users to use the prototype without detailed task to do or queries to send. This will record users interactions without constrains. In addition, if users have this freedom to search, they will try to test the system's functionality and try the system harder, so we can see if an error occurs or if everything works properly.

- The fourth task is to **define and develop adaptation techniques applied to the retrieval functionality of an IMIR system**. The adaptation will be based on user behavior, which is extracted from the past interactions. The set of interactions will be logged during the evaluation of the basic prototype. These interactions are recorded and analyzed to detect a number of patterns and rules for adapting the functionality of the retrieval. To extract patterns of interactions, we will use three semi-supervised machine learning techniques: decision trees, neural networks and clusters (see section 6.1).

- The fifth task of the thesis is the **evaluation of the adaptation techniques** (described in chapter 7), which has the goal of validating the performance of the implemented adaptation to the past interactions. It will be a comparative evaluation between every combination of machine learning (classification) algorithm, query features and scores of retrieval engine's rankings. This evaluation will be carried out without recruiting users. The same queries and relevance judgments done by users in the basic prototype will be used for this evaluation (see figure 1.4).

- The last (sixth) task is the **implementation of a health-domain prototype** (described in chapter 8), which has the goal of validating the adaptability of the model to a new and specific domain. Contrary to the sports-domain prototype, it is evaluated by means of its functionality, i.e., no users are recruited for evaluating the system. The evaluation is done by means of IR measurements.

1.6 Structure

This thesis is organized into 9 chapters which are enumerated next.

1. The **Introduction** presents the evolution of internet since its creation and describes the problem of multimodal information retrieval. Besides, it describes the motivation that has led us to investigate this problem. The generation of a multimodal retrieval system, as well as a formal model defining these systems and the adaptation of their functionality to user behavior are the goals of this thesis. In order to simplify the further reading of the thesis, it describes the terminology that explains how the elements are defined in each research area and highlights how we will refer to each component.

2. **Multimodal information retrieval** chapter reviews works related to multimodal information retrieval. Besides multimodal information collections, it also describes different types of multimodal queries. The retrieval engine selection module and the combination of results are also studied and analyzed. Later, the most relevant works are summarized, together with their advantages and disadvantages. Finally, the Federated Web Search Track 2013 [Demeester et al., 2013]

is described, with emphasis on the techniques used for source selection and results combination.

3. The first part of the proposal is fully described in the chapter **A Model to describe MIR systems**, which describes the formal model encompassing the definition of interactive multimodal information retrieval systems. This model includes every essential component of an IMIR system giving a formal and generalized (as much as possible) definition of them. These elements are: multimodal information organized in collections, (multimodal) query, different retrieval approaches, 'handler' and 'fusion' modules that are responsible for selecting the requested sources and the combination of results and user interactions.

4. Once the model is defined, the thesis proceeds with the implementation of a basic prototype based on this model: **Development of an IMIR prototype in sports domain**. The prototype implements a subset of the features of the model. The implemented features are: a multimodal collection that is semantically related (this collection has been generated for the Buscamedia project[29]), three types of multimodal queries (text, audio, and text + image), six different retrieval engines (question answering, full text search, ontology-based search, OCR in image, object detection in image and audio transcription), a fusion strategy based on expert-defined rules, a round-robin-based results' combination strategy and the registration of interactions.

5. The first part of the chapter **Analysis of the prototype functionality** encompasses the evaluation of the IR performance. It presents the methodology to validate and evaluate the prototype as well as the results obtained during this evaluation. The second part of the chapter focuses on validating how users have used the system by measuring the visualizations and capabilities of the system.

6. **Adapting IR functionality based on user interactions** presents the techniques that are used to adapt the functionality (performance) of the IMIR prototype based on past user interactions. Besides, three classification techniques are introduced: decision trees, multilayer perceptron and K-means. These techniques

[29]Buscamedia Project (see section 4.1) is a research project funded by the Spanish Ministry of Industry (http://www.cenitbuscamedia.es/))

are used for generating models which are trained using the past user interactions. Besides, it also introduces the information related to the query and the score used for ranking the REs as input for training the models.

7. **Experimental setups of IR adaptation based on user interactions**: this chapter shows the experiments carried out for validating the IR adaptation algorithms. Besides, the results obtained and a complete discussion about these results are included.

8. **Development of an IMIR prototype in health domain for social media analysis**: the thesis proceeds with the implementation of a prototype based on the model in the health domain. The goal of the prototype is the analysis of health social media streams in order to extract drugs, effects and their relations. The prototype serves as a first step for defined a complete multimodal retrieval system.

9. **Final remarks**: this chapter describes all the conclusions obtained from the results. Specially, the conclusions coming from the handler strategies and their evaluation are highlighted. It is pointed out that the performance of an IMIR system can be adapted using user-specific information, i.e past user interactions. It shows the impact that the work developed in this thesis has had a journal publication, conference publications and participation in research projects. Besides, a set of possible ways to continue or improve the presented work are described in the last part. Therefore, it also refers to a set of new applications that can benefit from the knowledge, methods and techniques generated or studied in this thesis.

2

Multimodal Information Retrieval

Interactive Multimodal IR (IMIR) is an issue that has been studied in depth. This chapter makes a review of the related work and it is organized based on the elements that compose an IMIR system. The information retrieval process is depicted in figure 2.1 where the main components are identified.

The whole process begins when the user has an **information need**. As an example we take a user that *'organizes a travel to Barcelona'*. In this case, the most relevant information for the user would be transportation to the destination (planes, trains, buses, private car), the required documentation if necessary, information about hotels or other places to stay (prices, geographic location, category, etc.) as well as the destination leisure and entertainment (clubs, restaurants, sights, places of tourist interest, etc.). This information need must be converted into a format that IR systems can understand through a process called **query formulation**. In many cases it is the user who performs the query formulation by selecting keywords that seem most appropriate to find the information needed. To meet this need for information the user may generate the following query: 'Barcelona', 'travel to Barcelona', 'hotels in Barcelona' or 'tourism Catalonia'.

At that time, the **query** is sent to the **matching algorithm**, that compares the query against the documents containing the information to be retrieved. In case of

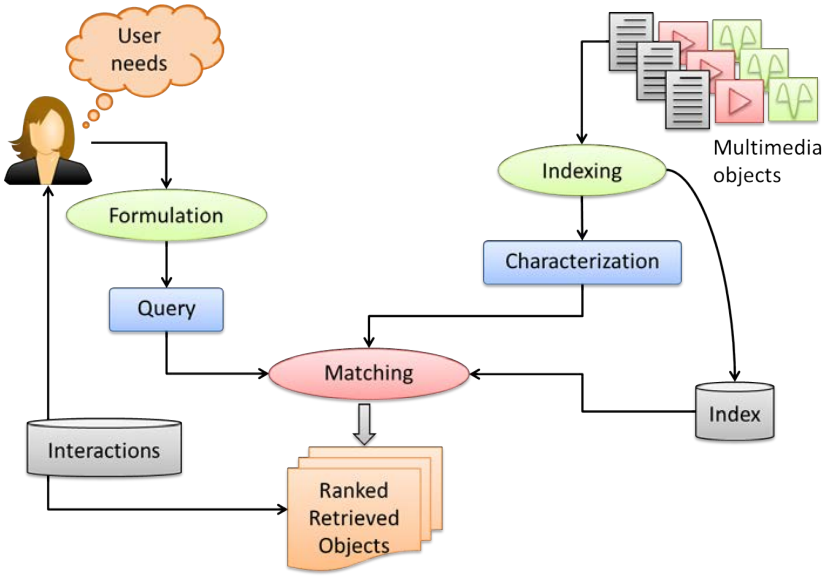


Figure 2.1: Abstract architecture of information retrieval (IR) systems

multimodal retrieval, these **objects** (texts, images, videos, etc) are requested to different retrieval engines. Each engine has previously processed the documents to extract relevant information from them. The extracted information is stored in a system of fast access to efficient retrieval. This system is called **index**. The index contains the representation of the information contained in both documents and metadata. The generation of this information and its inclusion in the index is the process known as **indexing**.

It is quite possible that the user does not want any purchasing information or scientific documents relating to Barcelona. Therefore it is important to distinguish between the most appropriate sources for each query. The current search systems request all existing sources without making any distinction. This strategy handling multiple retrieval engines (**Handler**) takes the decision of what retrieval engines are requested and in which way or order.

The matching between the query and the representation of documents returns to the user a set of **retrieved objects**. These objects are usually documents of the collections, but can also be parts of them, summaries or individual terms. When planning a trip (to Barcelona) is interesting to obtain all necessary information relative to the trip, not just pages that can give us information. It could be very useful to get a price

comparison table of transportation or hotel, a graphic with weather conditions during the travel dates, etc.

When several retrieval engines are requested, a set of results is obtained from each one. The **results fusion module** is in charge of combining them in order to get only one single final results' set. Currently most of commercial systems show the results (texts) in a single list, interspersed in some cases results of other so-called vertical systems (see section 2.3). These verticals are systems that retrieve documents from other modes (image or video) or other sources (purchases, blogs, etc).

In the example of organizing a trip to Barcelona, it is interesting to request *eDreams*[30], *Booking.com*[31], *Barcelona tourist office*[32], etc. and provide all the results of each system in a single list or adding information in a single view. In this way the user will have all the information you need to prepare for the trip only performing a search on a system without having to waste time while navigating through all the pages and compare all the results.

2.1 Multimedia Information

The first distinguishing characteristic of multimodal systems is the format of the collection they work with and how they handle this information. Information collections are divided according to the mode of the objects that compose it. A monomodal collection contains items from a single mode (text, image, video, audio, etc.). By contrast, a multimodal collection contains objects in different modes (text and image, image and video or whatever). In this division there is a special case: monomodal collections containing multimedia objects accompanied by metadata. Metadata is structured information accompanying multimedia objects. This metadata can be very diverse: the creator of the object or the location where it was created, the caption of an image, the transcription of a video or an audio, annotation of the existing concepts in the object, etc. We will consider these collections as multimodal ones. This classification is shown in figure 2.2.

(Multimodal) Retrieval systems can be classified according to the characteristics of the collection(s) of documents used to retrieve information. In this case we distinguish

[30] http://www.edreams.es/travel/?mktportal=EDR_ES accessed at 23/07/2015
[31] http://www.booking.com/index.es.html accessed at 23/07/2015
[32] http://www.turismedebarcelona.net/ accessed at 23/07/2015

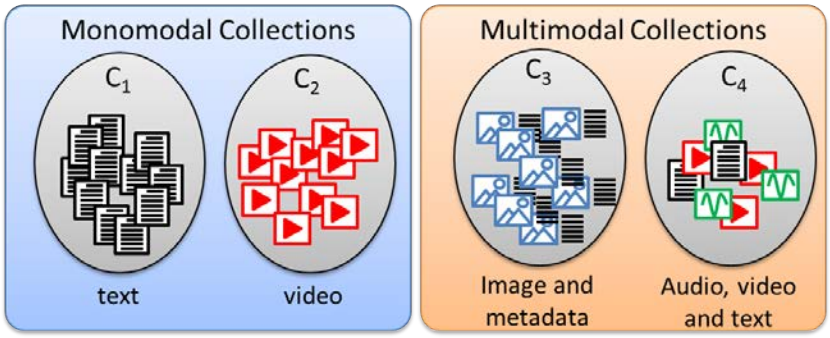


Figure 2.2: Classification of multimodal collections according to the mode of the documents composing them.

three types. The first type are the monomodal systems that work with a collection or a set of collections where only documents in one mode are included. The second type are the systems using several collections where each one contains only documents in one mode but at least two collections with different mode are used. The third type are the systems using a single multimodal collection including documents in several modes.

First we proceed to present a set of collections used for multimodal information retrieval. Then, a brief introduction to information management approaches used in multimodal information retrieval systems is done.

2.1.1 Collections

This presentation begins with some monomodal collections, and then continues to presents two types of multimedia collections: metadata-based and collections combining different modes.

Evaluation forums are scenarios in which systems participate for comparison with other systems in the same research area. These forums are interesting because they offer collections of documents that can be used, not only during the evaluation forum but also later, to benchmark systems easily.

There are several evaluation forums in information retrieval that are very interesting because over the years they have become a reference. These forums are the Text Retrieval Evaluation Conference (TREC), organized and held in North America, and the Cross-Language Evaluation Forum (CLEF), organized in Europe. These forums

are composed of evaluation tasks, each of which has a specific purpose within the world of information retrieval.

Monomodal collections

Currently there are works that use multimedia collections, but traditionally they worked only with textual collections to retrieve the information. This is due to the fact the text information in digital format has been available since the beginning of information retrieval, while multimedia elements are more recent.

The approach of Chernov et al. [2006] uses the ArXiv.org, HU-Berlin EDOC and CiteSeer OAI collections that contain metadata and full text. EDOC is an open access collection[33], which contains 2500 full-annotated research documents. ArXiv.org is also open access[34] and contains 1,024,344 e-prints documents in Physics, Mathematics, Computer Science and Biology. CiteSeer OAI comprises 750000 documents and is available under a creative common license[35] (CC BY-NC-SA 3.0). Another work using the CiteSeer collection is Golovchinsky and Diriye [2011]. Another text collection is WT10g test collection of the TREC 2001 Web Track [Hawking and Craswell, 2001] that is used in Buccio et al. [2010]. It contains 1 692 096 text documents[36].

In Hong and Si [2012] two TREC datasets besides of a Wikipedia dataset based on the ClueWeb are used. TREC7 and TREC8 Ad Hoc Tasks [Callan et al., 1992] are used in Shen and Zhai [2003]. These collections are composed by 1.5GB of text documents. It is not freely available but it can be bought[37].

A collection composed of news articles is used in Ahn et al. [2011]. It uses the Topic Detection and Tracking (TDT4) test collection available through an agreement signing[38] and encompassing 7430 broadcast news.

A collection of structured documents in XML format is used in Bessai-Mechmache and Alimazighi [2012]. It uses a subset of the collection of INEX (INitiative for the Evaluation of XML retrieval), which contains 144,625 documents. This collection can be download[39] if a form is submitted. Besides, there are works that retrieve textual

[33] http://edoc.hu-berlin.de/ accessed at 23/07/2015
[34] http://arxiv.org/ accessed at 23/07/2015
[35] http://citeseerx.ist.psu.edu/ accessed at 23/07/2015
[36] `http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html` accessed at 23/07/2015
[37] `http://trec.nist.gov/data/test_coll.html` accessed at 23/07/2015
[38] http://ssli.ee.washington.edu/people/leixin/TDT4.html accessed at 23/07/2015
[39] http://www.inex.otago.ac.nz/ accessed at 23/07/2015

elements instead of complete textual documents: such as Balog et al. [2012]. This work uses the Billion Triple Challenge 2009 (BTC-2009) dataset. This collection is composed by 1,464,829,200 RDF (resource description framework) statements describing entities and is open access[40]. These entities are the retrieved information.

Multimodal collections (multimedia with metadata)

Most systems working with multimodal collections manage only two modes, one being usually textual information. Metadata-based retrieval obtains documents whose associated data are similar to the query. This information is considered as two separated collections: multimedia elements are seen as one collection and is requested by their content (Content-Base Information Retrieval - CBIR). Their associated metadata are considered as other collection.

A search on the metadata associated with images is performed in Lana-Serrano et al. [2011] that retrieves image and text using the ImageCLEF 2011 Medical Retrieval Task dataset [Kalpathy-Cramer et al., 2011]. This dataset encompasses 77000 images and is available[41] after asking for access. Another work using this collection is Caicedo [2009] that performs image retrieval.

The work of Romberg et al. [2012] mixes images and text. They created an image dataset called Flickr-10M available upon request[42]. It is composed of 10 million images downloaded from Flickr. Another work mixing images and text is Wang and Smeaton [2012] that retrieves images and concepts associated to them using a collection their recruited using SenseCam [Hodges et al., 2006].

Demner-Fushman et al. [2012] combines **text and visual features** (images features) in document representations. In the biomedical domain it accesses a collection comprising 600k images and 250k medical articles. Images together with text from the ImageCLEF2010 Wikipedia collection are managed in Arampatzis et al. [2011]. This collection contains 237,000 Wikipedia images and is available for download[43] by obtaining an account. A collection of 2500 images with accompanying texts is managed in Srihari et al. [2000] while Kludas and Marchand-Maillet [2011] uses a subset of the Corel dataset which contains keyword annotated photos.

[40] https://km.aifb.kit.edu/projects/btc-2010/ accessed at 23/07/2015
[41] http://www.imageclef.org/2010/medical
[42] http://www.multimedia-computing.de/wiki/Flickr-10M accessed at 23/07/2015
[43] http://www.imageclef.org/wikidata accessed at 23/07/2015

Although the management of **text and music** is not so common, Hu et al. [2011] uses a collection composed of 750 songs from U.S. and U.K. popular music.

There are also works that manage **video and text**. Yilmaz et al. [2012] uses TRECVID[44] 2007 collection that includes 100 hours of multilingual video and annotations. It is composed by news magazine, science news, news reports, documentaries, educational programming, and archival video and BBC archive files. This collection is publicly available[45] but a consent must filled.

Multimodal collections (integrating different modes)

Those studies that use more than two formats are not very frequent but also exist. An example is Marchand-Maillet et al. [2011], that can handle each multimedia mode. It uses a subset (*'1000 images'*) of the Corel image collection[46] comprising *'68040 images'* together with annotations.

The work of Jou et al. [2013] has created its own multimodal collection encompassing documents of three types: 18000 hours of broadcast news, 3.58 millions of articles news taken from Google News[47] and 430 millions public messages from Twitter[48]. It crawls the three sources at the same time in order to obtain topic-related documents.

It is interesting to highlight that a new evaluation forum called Federated Web Search Track[49] (taking place together with TREC) has consolidated. It has been celebrated in 2012 and 2013 and one of its goals is to evaluate and compare different resources selection and results' combination strategies in federated search. For tests it uses a collection of searches and results for a set of 157 search engines. Although this collection contains only text (webpages), there are multimedia elements contained in them. It is composed by 1,894,463 web pages. This collection is publicly available[50] (2012 and 2013 datasets) while the latest release (2014 dataset) is only available if taking part in the track. A complete description of the forum is done in section 2.8.

[44] http://trecvid.nist.gov/ accessed at 23/07/2015
[45] http://trecvid.nist.gov/trecvid.data.html#tv07 accessed at 23/07/2015
[46] https://archive.ics.uci.edu/ml/datasets/Corel+Image+Features accessed at 23/07/2015
[47] https://news.google.com/?hl=en accessed at 23/07/2015
[48] https://twitter.com/?lang=es accessed at 23/07/2015
[49] https://sites.google.com/site/trecfedweb/ accessed at 23/07/2015
[50] https://sites.google.com/site/trecfedweb/2013-track#obtaining-collection accessed at 23/07/2015

Collections Summary

The properties of the collections are summarized in table 2.1. Each column stores a properties of the collection: the **mode** of the objects it contains is stored through their first letter (audio-A, image-I, Text-T, video-V and metadata-M), the **language** and the **size** of the collection, the **organization** of the documents and the collection (storage way, format, etc.), information about the **evaluation forum** in which the collection is used (if done), the availability (**Avail.**) of the collection (if a license is needed) and the **URL** where it can be found. Whenever an information is not known it is marked as ***NK*** and if the information is not available a ***NA*** is used. Each collection is described (together with its properties) in a separate row.

Table 2.1: Comparison of multimodal datasets

	Mode	Language	Size	Organization	Evaluation Forum	Available	URL
ArXiv.org	T	EN	1,024,344 e-prints documents		NA	Y (Open access)	http://arxiv.org/
HU-Berlin EDOC	T	EN-DE	2500 full-annotated research documents		NA	Y (Open access)	http://edoc.hu-berlin.de/
CiteSeer OAI	T	EN	750000 documents		NA	(CC BY-NC-SA 3.0)	citeseerx.ist.psu.edu/
WT10g	T	EN	1 692 096 documents		TREC 2001 Web Track	Sample (Sign agreement and pay)	http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html
TREC7 and TREC8 Ad Hoc Tasks	T	EN	1.5GB of documents	TREC 7 and TREC 8	Used in Shen and Zhai [2003]	N (90$ per disk)	http://trec.nist.gov/data/test_coll.html
TDT4	A-T	EN, AR, CH	7430 Broadcast news audio	Used in Ahn et al. [2011]	Topic Detection and Tracking 2002 and 2003	N (Agreement)	http://ssli.ee.washington.edu/people/leixin/TDT4.html
INEX XML Subset	T		144,625 documents	Used in Bessai-Mechmache and Alimazighi [2012]	INEX 2010 Tracks	N (Login required)	http://www.inex.otago.ac.nz/
Billion Triple Challenge 2009 (BTC-2009) dataset	T	EN?	1,464,829,200 RDF statements	Used in Balog et al. [2012]	TREC 2010 Entity Track	Y (Open access)	https://km.aifb.kit.edu/projects/btc-2010/
ImageCLEF 2011 Medical Retrieval Task dataset	T-I	EN?	77000 images	Used in Lana-Serrano et al. [2011]	ImageCLEF 2011	Y (Login required)	http://www.imageclef.org/2010/medical
ImageCLEF2010 Wikipedia collection	I-M(T)	EN, DE, FR	237,000 Wikipedia images	Used in Arampatzis et al. [2011]	ImageCLEF 2010	Y (Login required)	http://www.imageclef.org/wikidata
Flickr-10M	T-I	All[51]	10 million images downloaded from Flickr	Used in Romberg et al. [2012]	NK	Y (Upon request)	http://www.multimedia-computing.de/wiki/Flickr-10M
Demner-Fushman et al. [2012]	T-I	EN	600k images and 250k medical articles	Used in Demner-Fushman et al. [2012]	N	N	NA
Ad-hoc [Jou et al., 2013]	T-V-T	EN	18k hours broadcast news, 3.58M articles news and 430M Twitter messages	NA	N	N	NA
Ad-hoc collection	A(Music) - M(T)	EN	750 songs	U.S. and U.K. popular music	N	N	NA

[51]There was no filtering when downloading images so there is no limitation in language

TRECVID 2007	V-M(T)	EN	100 hours of multilingual video and annotations	News magazine, science news, news reports, documentaries, educational programming, and archival videoand BBC Archive	TRECVID 2007	Y (Form required)	http://trecvid.nist.gov/trecvid.data.html#tv07
Corel image collection	I-M(T)	EN?	68,040 images	Used in Marchand-Maillet et al. [2011]	NK	NK (Maybe upon request)	https://archive.ics.uci.edu/ml/datasets/Corel+Image+Features
Federated Web Search Track	Every[52]	NK	1,894,463 web pages	NK	Federated Web Search Track 2012, 2013 and 2014	Y[53]	https://sites.google.com/site/trecfedweb/

[52]Web pages contain text, but they are linking multimedia content
[53]Latest release is only available if taking part in the Track.

Our work concerns retrieval of multimodal information, so we will need a collection that includes as many modes as possible. Failing that, we could use several collections in different modes: one textual, one image and one video for example. Besides the different modes, an important feature of the collection (or collections) is that it must be accompanied by semantic information. This semantic information must not only annotate documents, but also must relate them. The relationships between documents must allow for exploratory navigation such as web links.

Most collections are not worth because they only contain documents in one mode or two, where the second mode is (textual) metadata. These collections (using only documents of one or two modes) are not useful for our purposes. In order to verify the development of this thesis we need as many modes as possible. This is due to the fact that all modes should be handled simultaneously in this thesis. Regarding multimodal collections (more than two modes), we found only collection of Federated Web Search Track (see section 2.8), which has the limitation that no semantic information (linking documents) is available. FedWeb suffers the problem that the retrieved information is not semantically related. This is a clear disadvantage for the purposes of this thesis.

2.1.2 Collections Management

A system can use monomodal or multimodal collections, but it is not only distinguished by that, but we can also classify systems depending on the way they handle or manage the information. This information management is done mainly in two ways: (i) the multimedia objects are handled as they are, i.e without specific representations; and (ii) the multimedia objects are transformed to represent them by specific languages or to include them into joint indexes.

Most multimodal information retrieval works do not perform a specific representation of multimedia objects, but use them in their original format. Every work performing content-based retrieval (CBR) manage documents in their original format.

In this type of information management we include all those systems that retrieve information from web search systems. These systems do not represent the information, but they function as mere intermediaries, sending requests to every available vertical[54] and returning the results. Some of them do a post-processing to the results, but do not change its representation, but can change their order or grouping.

[54]See footnote 23.

Available internet search engines have also been used as retrieval engines. Systems like Yahoo or Bing (plus their verticals) are used in different works. Malla et al. [2011] uses Bing as engine and all its verticals: maps, news, images and advanced search. Bing, Google, Yahoo and Ebay are used in Arguello et al. [2012] to perform aggregated search using their verticals. It uses the next document modes: images, videos, news, blogs, community Q&A and shopping. In Renaud and Azzopardi [2012] is introduced a system that can use the application programming interfaces (APIs) of a large number of search engines[55] to get results from them.

Works that make a representation of the multimedia elements using a specific language must also be considered. Documents of all modes are managed in Nottelmann and Fuhr [2003]. It presents the transformation of documents into DAML + OIL [Frank van Harmelen and Peter F. Patel-Schneider and Ian Horrocks, 2001], which is a RDF-based language that enriches RDF with more advanced primitives and allows the representation of multimodal documents. Another work that uses its own resources description languages is Steiner et al. [2012]. Rich Unified Content Description (RUCoD) [Daras et al., 2011] represents the multimodal documents. It is a XML-based language. The modes it accepts are: audio, video, image, emotion, geolocalization and text.

An important point of multimodal retrieval is the systems that create combined or centralized indexes containing all modes of documents. A work in this line is de Vries [1998] using low-level features of multimedia elements (audio, video, image, text, etc.) to integrate them into an *'open distributed architecture'*, i.e. a multimedia database storing multimedia data and its associated meta-data.

Those studies that use more than two formats are not very frequent but also exist. A clear example of such systems is Yang et al. [2002], where the multimodal system Octopus is presented. It allows the retrieval of multimodal documents that are stored in an integrated database. Besides, it also makes use of the Multifaceted Knowledge Base (MKB) that is a three layer network containing nodes (N) and links (L). The first layer (*feature layer*) refers to the low level features of media objects. The second layer (*structure layer*) represents structural relationships among objects. The third layer (*perception layer*) manages user relevance among objects.

Another example is Marchand-Maillet et al. [2011], that defines a matrix representation where the documents are represented by means of a matrix. Position (i, j) in the

[55] Bing, Twitter, YouTube, iTunes, Wikipedia, Picassa, Flickr, and Digg

matrix stores the value of feature j of documents i. Every multimedia mode (audio, video, image, text, etc.) can be represented in this matrix.

As far as collections management is concerned, the use of web search engines affords the problem that the retrieved information is not semantically related. The same problem applies to the collection created by FedWeb. Systems that use specific representation languages (DAML&OIL [Nottelmann and Fuhr, 2003] and RUCoD [Steiner et al., 2012]) are interesting due to the fact that they handle every possible information mode. Works using joint indexes [de Vries, 1998; Marchand-Maillet et al., 2011; Sushmita, 2012; Yang et al., 2002] are remarkable. Big data is becoming a trend and techniques handling big data have increased efficiency and effectiveness. Despite, joint (or combined) indexes can have a scalability problem if they increase their size too much (web size can be considered as big data).

2.2 Information need representation: Query

The query is the system-understandable representation of the user information need. It is important that queries represent adequately the information need to resolve it as efficiently as possible. This makes the use of multimodal queries (with multimedia elements) more appropriate when we are looking for multimodal content (images, videos, etc.).

The query mode (text, image, text and video combined, etc.) that is accepted in these systems is important, as well as the way the query is represented. Figure 2.3 displays several examples of queries. Three monomodal queries are shown top left: a question ('Who is the president of UEFA?'), a keyword query ('Biggest river in Thailand') and an image (used in image CBIR systems for example). Top right shows a multimodal query, where an image and a text are combined. Below a query represented by a language-specific representation (SPARQL) is shown. This query asks for every instance of a knowledge system (an Ontology for example) that *shows* the concept *Fernando Alonso*.

2.2.1 Monomodal queries

In most cases, the query arises in textual mode, perhaps because from the start systems admit this type of query and the user is used to them. Therefore textual query con-

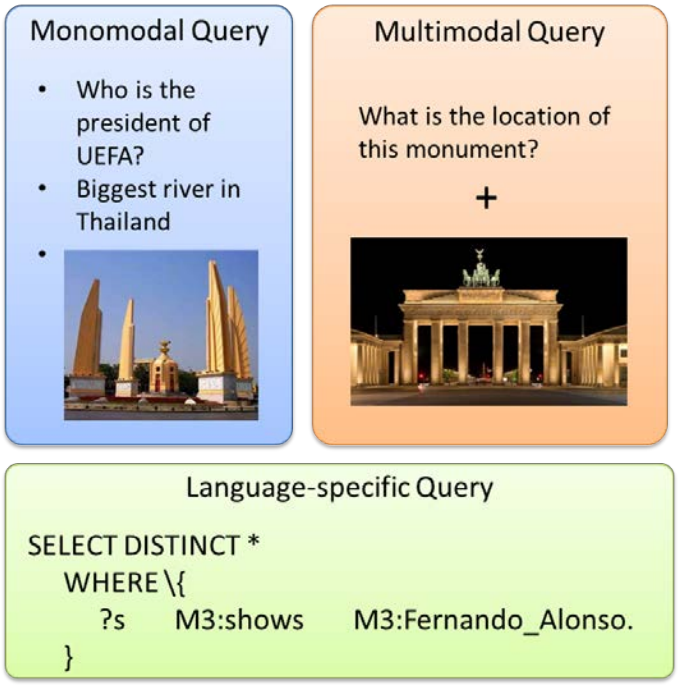


Figure 2.3: Examples of multimodal queries. Three monomodal queries are shown top left: a question ('Who is the president of UEFA?'), a keyword query ('Biggest river in Thailand') and an image (used in image CBIR systems for example). Top right shows a multimodal query, where an image and a text are combined. Below a query represented by a language-specific representation (SPARQL) is shown. This query asks for every instance of a knowledge system (an Ontology for example) that *shows* the concept *Fernando Alonso*.

struction is simple and intuitive for users to express their information needs. Table 2.2 describes three examples of monomodal queries together with their associated information need. There are plenty of works that use keyword as textual queries such as Ahn et al. [2011], Hong and Si [2012], Beckers and Fuhr [2010] or Görg et al. [2010]. Other works using keyword-based queries are those that retrieve information using commercial internet search engines (Yahoo!, Bing, Google, etc.) such as Sushmita [2012], Malla et al. [2011] or Arguello et al. [2012].

There are works that use textual queries to retrieve multimedia elements by matching the query content (text) against their associated metadata. Some examples are Caicedo [2009] that performs image retrieval starting with keywords, Vallet et al. [2012] retrieving videos or Hu et al. [2011] using a two-step retrieval strategy for music.

Query	Information Need Description
Who is the president of UEFA?	A user wants to get the name of the president of UEFA institution. As it is not specified, (s)he can also be interested in previous president, not just the current one.
Biggest river in Thailand	A user who will visit Thailand soon is interested in information about a river that (s)he will navigate in.
videos goals Manchester United	A fan of the football team wants to watch videos of the goals of Manchester United footbal team.

Table 2.2: Examples of monomodal queries together with the description of its associated information need

Some works use multimedia elements as queries, although they are less common. Table 2.3 displays two queries containing a single multimedia element and their associated information need. Image queries are accepted in Wong et al. [2005] and Suditu and Fleuret [2011] that perform image retrieval based on low-level features. Interesting is Hauptmann et al. [2002] that allows voice queries to make metadata video retrieval. In Yang et al. [2012] pictures or short video can be used as query.

Query	Information Need Description
	A user wants to get information about the monument Brandenburg Door, but not only the textual information such as creation or interesting facts. (S)he could be interested in similar images, a map offering its location, etc.
File containing a song of Metallica	A user wants to hear music similar to the song (s)he is send to the system. Besides, (s)he could find interesting to obtain related information to the band playing the song.

Table 2.3: Examples of multimodal queries together with the description of its associated information need

2.2.2 Using more than one mode: multimodal queries

The elements that compose a (multimodal) query are the third property that characterizes IMIR systems. When using multimedia elements it is complicated to select the right elements that represent user information needs to compose a query. For this reason, the decision of how many elements to use (and their type) depends on the purpose of the research.

A wide variety of works use textual query in conjunction with a multimedia element. The most common combination is text and image such as Demner-Fushman et al. [2012] that allows monomodal (text or image) and hybrid query. When an image is sent as query (alone or together with text), it is processed to extract concepts and labels (text) present in the image in order to use them as search terms (together with the text query if present). Malla et al. [2011] is other example of image and text query. These modes are not combined, but they are used separately. A user can send a text or an image as query, that are represented by its content.

There are not many works that manage fully multimodal queries (considering more than two modes). Some work studied multimodal query representation by specific languages that allow identification of any media item. In [Nottelmann and Fuhr, 2003], which describes the MIND architecture, it is done a transformation of queries into DAML&OIL [Frank van Harmelen and Peter F. Patel-Schneider and Ian Horrocks, 2001]. This language can represent every multimedia element. An example of DAML&OIL represented query is shown in figure 2.4. The Rich Unified Content Description [Daras et al., 2011] also represents multimodal elements into the query. The modes it accepts are: audio, video, image, emotion, geolocalization and text.

An important point of multimodal retrieval are the systems that create combined or centralized indexes containing all modes of documents, representing the query in the same space of features to later make a matching between them. This kind of works accept multimodal queries containing elements in every mode. An examples is Marchand-Maillet et al. [2011] accepting media elements as query. This system is meant for collection mining (searches made for exploring the collection), therefore it only accepts media elements from the collection it is working with.

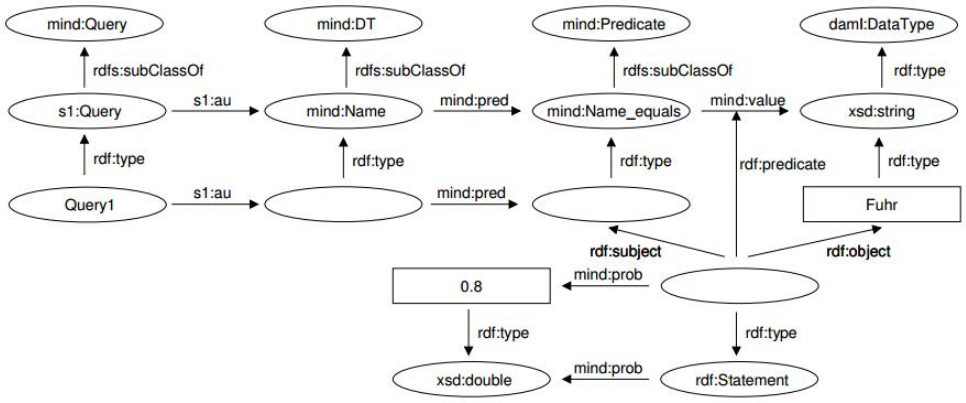


Figure 2.4: Example of query represented with DAML&OIL language taken from Nottelmann and Fuhr [2003].

Another example is de Vries [1998], which uses a query processing system. This system generates a suitable query for a multimedia database, which depends on two things: the query generated by the user and historical (previously performed) interactions performed on the results. It represents the multimedia query using model object algebra. An example of the representation of a multimedia element (video) is shown in table 2.4 , where *Time*, *Date*, *Image*, *Audio* and *Text* are simple atomic data types.

```
BAG<
 TUPLE<
time:Atomic<Time>,
date:Atomic<Date>,
keyframes:LIST<
 Atomic<Image>
>,
audiotrack:Atomic<Audio>,
transcript:Atomic<Text>
 >
>;
```

Table 2.4: Example of query representing a video in de Vries [1998].

By contrast, multimodal elements are used in Srihari et al. [2000] as query. The query contains a set of components, that are low-level features: text strings, image fea-

tures, object categories, spatial relationships or metadata. An example of multimodal query is shown in figure 2.5.

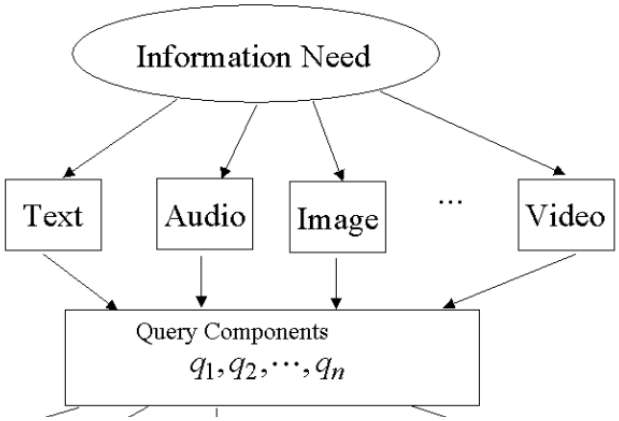


Figure 2.5: Graphical example of multimodal query in the work of Srihari et al. [2000].

Another work accepting multimodal queries is Yang et al. [2002], whose system allows the use of multimodal elements the same way as query-by-example: media elements are represented by their content, not their features.

Monomodal queries are not worth because we do not have enough expressiveness. As claimed in de Vries [1998], *'textual queries cannot capture the full semantics of multimedia data'*. Multimedia queries are mainly limited to contain element of two modes. It would be interesting not to limit the number of modes in a query. For this reason, we are going to borrow the approaches of Yang et al. [2002] and Marchand-Maillet et al. [2011]. We will focus on those approaches to define and design the management of multimodal query.

2.3 Retrieval Techniques

Information retrieval is defined as obtaining relevant documents to fill a user information need. To address this need, the user formulates the query in an understandable way for an automated system that will compare the user generated query with every document it contains. This comparison aims to find all documents related to the user's query. This process is called matching, and can be classified into three types: content-based retrieval, metadata-based retrieval and other approaches. This section

will present some techniques for matching between query and documents used in the related literature.

Content-based retrieval compares the content of the query to the content of documents. When the content is textual, the documents are represented using known models such as probabilistic model [Jones et al., 2000], vector space model [Salton et al., 1975] or boolean model [Cavanagh, 1976] in order to compare them, while multimedia elements are analyzed to extract features, called *low-level features*. These features are the elements being compared between the query and the documents, i.e the same features are extracted from the query and documents and a comparison is performed on them. For metadata-based retrieval, documents are searched by content, but also for its associated metadata. This metadata is structured information associated with multimedia elements. Examples of metadata include: person, time or place of creation of the media object, information about the content (objects in images, speakers on audio, etc.). The third type encompass all those approaches that are not classified in the first two types. For example, systems that perform a representation of multimedia objects using a specific language (DAML&OIL, XML, RDF, etc.).

Figure 2.6 displays a classification of several retrieval engine techniques. Left part includes two typical CBIR searches: one being a text-based search and the other a multimedia low-level feature-based retrieval. Center shows a metadata-based search where multimedia elements (together with their metadata) are returned by matching text against multimedia element's metadata. Right part shows a joint index retrieval approach, where documents of every mode are combined and queries of every mode are used for requesting.

2.3.1 Retrieving documents by its content: Content-Based Information Retrieval (CBIR)

Content-based information retrieval compares the contents of the query to the content of each document of the collection. If the mode is text, the comparison is performed by the text. If multimedia elements (images, videos, audios) are compared, the comparison is performed by the features of the media, which can low-level or high-level features. The low-level features are those that are drawn from the content of the multimedia object, such as diagrams color or texture in images, frequency analysis in audio or motion

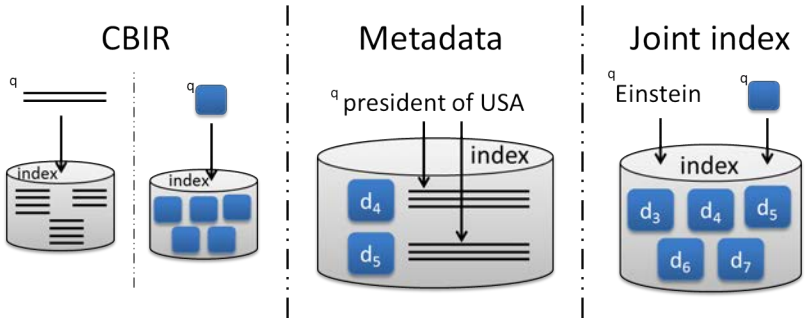


Figure 2.6: a classification of several retrieval engine techniques. Left part includes two typical CBIR searches: one being a text-based search and the other a multimedia low-level feature-based retrieval. Center shows a metadata-based search where multimedia elements (together with their metadata) are returned by matching text against multimedia element's metadata. Right part shows a joint index retrieval approach, where documents of every mode are combined and queries of every mode are used for requesting.

analysis in video. By contrast, the high-level features are the conceptual content of multimedia element, i.e., concepts that are present in the media item.

Retrieval through the matching of textual query and document content (keywords) is the most commonly used. Examples which may be mentioned are: Hong and Si [2012], Beckers and Fuhr [2010] (using Apache Solr[56]) or Görg et al. [2010].

Ahn et al. [2011] makes textual retrieval using TaskSieve [Ahn et al., 2008] and VIBE [Ahn and Btusilovsky, 2009] systems. They are adaptive exploratory search systems (ESS) which adapt their functionality to user models. In this work the patterns (for adapting functionality) are extracted to create user models in order to use them together with REs. Patterns are extracted using two approaches: overview of temporal sequences and alignment of specific sequences of interactions.

In other recovery modes each approach has its own characteristics. In Romberg et al. [2012], image retrieval based on low-level features (color, texture, shapes) uses mm-pLSA (multilayer multimodal probabilistic Latent Semantic Analysis) [Hofmann, 2001]. It generates a matrix where rows are images and columns are the low-level features in the image. The retrieval is done using this matrix. It also applies pLSA to the image tags. Another works that perform image retrieval based on low-level features are Wong et al. [2005] and Suditu and Fleuret [2011].

[56]http://lucene.apache.org/solr/ accessed at 23/07/2015

The social image retrieval engine (SIRE) is presented in Hoi and Wu [2011] for retrieving images. It has three search capabilities: text search by keyword using image title and annotated tags, visual search with low-level features (grid color movement, local binary pattern and gabor and edge features) and multimodal combination of both in a sequential two step approach: first a text-based query (keywords) is used to retrieve images, then the search can be refined by selecting some image from the results.

Another interesting retrieval approach is presented in Hauptmann et al. [2002], which allows video retrieval by searching into metadata that have been previously extracted: OCR (clustered sharp edges using horizontal diferential filtering [Sato et al., 1998]) and speech recognition (Sphynx [Singh et al., 2001]). The matching implements the OKAPI algorithm [Robertson et al., 1992].

Vallet et al. [2012] describes another video retrieval system, but it uses external knowledge sources to make an iterative search. First it searches for images in external sources with text: DBPedia[57], Flicker[58] y Google images[59] and then these images are used for retrieving videos by visual content.

Available internet search engines have also been used as retrieval engines. Systems like Yahoo or Bing (plus the entire verticals[60] they offer) are used in Sushmita [2012], Liu et al. [2011], Malla et al. [2011] or Arguello et al. [2012]. The approaches presented in these works use web search engines as retrieval engines, so they only act as proxies which transport the query and the results between the user and the web search engine.

2.3.2 Associated information retrieval: Metadata-Based Information Retrieval (MBIR)

Metadata-based search systems use not only the content of the multimedia element, but also its associated metadata to retrieve relevant documents to a query. As said above, metadata is structured information associated with multimedia elements. There are many different types of metadata ranging from multimedia item generation annotations (geolocalization, person, etc.) to more complex information such as semantic concepts identified within the item. Since much of the information associated with multimedia

[57] http://es.dbpedia.org/ accessed at 23/07/2015

[58] https://www.flickr.com/

[59] https://images.google.com/ accessed at 23/07/2015

[60] Vertical is more used for different retrieval engines inside a portal such as Yahoo! and its verticals: web, images, maps, blogs, etc. Each of these retrieval engines is known as a vertical.

elements is text, this type of search starts or contains a textual search in most of the cases.

A search on the metadata associated with images is performed in Lana-Serrano et al. [2011]. First it retrieves images according to search the caption of the images and then on those images it performs CBIR. The system is divided into five modules: query expander (adds related terms), textual retrieval using Lucene [McCandless et al., 2010] index, visual retrieval using LIRE [Lux, 2011], visual classifier (decides which class images belong to) and results combination.

Another interesting work is Torres [2005], which defines the Visual object information retrieval (VOIR) prototype, which combines two layers (conceptual and feature-based) to perform retrieval. It accepts text-based queries and returns videos. The retrieval is performed using the videos associated information (extracted in the process of indexing by performing automatic transcription and key frames extraction). The concepts extracted from keyframes and text are taken from a textual thesaurus.

The system of Wang and Smeaton [2012] works together with images and text. It retrieves images and concepts associated to them. It uses a semantic space of concepts, event semantic space (ESS), to group them and to accomplish concept-based retrieval. An ontology is used to represent the concepts and concept relations within domain, that are clustered according to their distribution in the semantic space.

Text and music retrieval is performed in Hu et al. [2011] that uses a two-step retrieval strategy: it searches for similar audio elements using text and then requests a content-based music retrieval system. It uses the *Mooddydb* system that performs music mood classification and retrieval. This system handles salient spectral features and sequential minimal optimization [Platt, 1999] (SMO). The search starts by artist or title of song (text) and then a selected song is used as seed for SMO.

Demner-Fushman et al. [2012] defines the OpenI multimodal IR prototype. It uses a multimodal index where textual content and image low-level features are added. There are two processing pipelines one for text and one for images. Text is indexed using a domain specific search engine Essie [Ide et al., 2007] and Lucene [McCandless et al., 2010]. For image, the considered image low-level features are: color layout description (CLD), color coherence vector (CCV), edge histogram description (EHD), discrete wavelet transform (DWT), average gray level (AGL), color edge direction descriptor

(CEDD) and fuzzy color texture histogram (FCTH). This features are matched using Lucene image retrieval engine [Lux, 2011] (LIRE).

An important point of multimodal retrieval are the systems that create combined or centralized indexes containing all modes of documents, representing the query in the same space of features to later make a matching between them. Marchand-Maillet et al. [2011] defines a unified model where it aggregates documents, concepts and users in a *'multi-tripartite graph'*. Documents are represented in a matrix containing relations between documents. Another matrix represents concepts and another matrix contains tags (relations between docs and concepts). There are other two matrix: one of users who represents the social network and another that determines which documents have been created or rated by each user. The combination of these matrix conform the *'multi-tripartite graph'*. It performs exploraty search, i.e., only documents of the (included) collections can be used as query, so the retrieval is done analyzing the links and relations inside the multi-tripartite graph.

Octopus [Yang et al., 2002] is a multimodal retrieval system that represents multimodal information in a single index (named as Multifaceted Knowledge Base - MKB), which models different levels of knowledge and relevance between media elements. Octopus uses a specific retrieval approach defined as Link Analysis based retrieval (LAbR): it analyzes links inside MKB in two ways: analyzes the relations between documents from the same knowledge level and analyzes relations between knowledge levels. Then, it retrieves multimodal documents based on these links (relations).

Systems using joint index are more intuitive to manage multimodal documents, although those systems requesting several retrieval systems are more useful due to the current online information distribution. For using joint indexes, the whole available information must be combined together to generate the indexes. It is a hard task when it comes to talk about web-size information. Systems using several REs present two advantages: (i) each RE offers its own search engine, which is optimized for searching on its information; and (ii) dividing information among several REs avoids scalability problems.

To add a new documents collection to the first type, the documents must be converted to the format of the index. For systems that request multiple systems, it is simpler: the new collection should be added to a RE and the RE should be consid-

ered within the system. Iterative systems are easily scalable, but too many systems iteratively queried can add a temporal constraint.

2.4 How to combine different retrieval engines?

In case several REs are requested, the system has to make them work together and to combine them. Furthermore, every engine will not be useful for every query, so the system has to decide which one will be requested in each search. Therefore, the retrieval engine selection or handling approach is in charge of selecting which retrieval engines are triggered by each query (in case there are more than one RE suitable for the input query) and in which order they are requested.

The basic handling approach that is analyzed is having no strategy, i.e. the query is sent to all available systems without distinction. Systems that typically use this approach are those that work with web search engines and their verticals. Sushmita [2012] does web searches on Yahoo! and some of its verticals: maps, blogs, and more, but it does not distinguish which of them are requested. Two other studies in this line are Arguello et al. [2012] and Malla et al. [2011], that use Bing (a commercial web search engine[61]) as engine as well as some of its verticals. Another work that makes no distinction between engines is Hong and Si [2012]. It accepts a text query and sends it to every available *RE* that accepts text as input.

On the other hand, there are some works that make distinctions in the Retrieval Engines (REs) which are requested. The RE selection criteria are very different from one work to another.

Another approach is dividing the query elements according to their mode and sending each element to the corresponding REs (every RE that accepts this mode at the input). This approach is used in Renaud and Azzopardi [2012], Demner-Fushman et al. [2012] and Romberg et al. [2012].

Some more complex techniques are observed in Chernov et al. [2006] that implements a *broker* (handler) to select the systems to be sent depending on the terms present in the query. This approach stores a resource description for each RE, which contains selected terms of their documents as a summary. A matching between the query and the resource descriptions of every RE determines which REs are requested.

[61] https://www.bing.com/?setlang=es accessed at 23/07/2015

A probabilistic approach to select REs depending on the relevant entities that each engine would return is presented in Balog et al. [2012]. It stores a summary of each collection and uses three approaches to determine the order of sources and the number of relevant results of each engine: (i) a collection-centric (CC) approach where each collection is considered as a single document and it is matched against the query; (ii) a document-centric (DC) approach where the score of a collection is computed by adding each score of matching the query and a document; and (iii) a linear combination of the two previous approaches named *All that an Entity Needs is a Name* (AENN).

Using multiple retrieval engines sequentially is a widespread approach, especially for metadata-based retrieval: first the textual query is matched with the meta-data of media elements, and then (in some cases) results from the textual search are used to query CBIR systems. The source selection strategy of this work is the order in which they are queried and the query that is used in later systems.

In Lana-Serrano et al. [2011] a sequential approach is used: first it uses a textual retrieval and then applies a visual retrieval over the results. Another work is Hoi and Wu [2011] that retrieves images in a two step approach: first text then image. A music retrieval system [Hu et al., 2011] searches for similar audio elements using text and then requests a content-based music retrieval system.

The approach proposed in Vallet et al. [2012] first searches with text in external sources: DBPedia (in Spanish it encompasses 100 million RDF triplets extracted from Wikipedia), Flicker (social network for photos and videos sharing) and Google images (web search tool for images) and then uses these images to retrieve video by visual content.

Torres [2005] describes a video retrieval system that returns videos and associated metadata using text queries. It explores associations between text (query) and image regions (through a textual thesaurus). Images are later used for requesting videos. Another video retrieval system is Hauptmann et al. [2002] that allows voice queries. It transcribes the query and matches it against the extracted information from videos: optical character recognition (OCR) and speech recognition using Sphynx [Singh et al., 2001] are used.

A speaker and topic recognition information retrieval system is presented in Jou et al. [2013]. This system first analyzes videos extracting topics and entities that are mentioned in videos. Then, it performs face recognition and speech segmentation.

Finally, it combines the extracted information for offering a results will the required information about speakers and topic altogether.

The work of Arampatzis et al. [2011] is interesting because it compares two approaches: fusion vs 2-stage process. The fusion strategy is based on a linear combination of two scores: a text retrieval score , Term Frequency - Inverse Document Frequency (TF-IDF), and a visual feature score ($DESC_{ij}$), which is the value of a low-level feature descriptor of an example image (of the collection it uses). The second approach is a sequential reordering approach, which re-ranked the k-tops results obtained by text query using the visual feature scores.

One of the goals of Federated Web Search Track (see section 2.8) is to evaluate and compare different resources selection strategies in federated search. In the resources selection task participants have to classify 157 engines for each topic without having access to the corresponding results. The participants must extract a set of retrieval engines. We describe here the best systems participating in the task that submitted any run to the source selection task. These systems are explained because the approaches they used inspire the strategies that we will develop inside the handler proposed in this thesis to manage the retrieval engines that are requested.

In Pal and Mitra [2013] search engines (SE) are ranked based on a score computed using the frequency of query terms present in the eight top results offered by each search engine. This score is computed using the term frequency of token q_i in document d.

A level-based approach was adopted in Buccio et al. [2013]. They defined three levels: term level, document level and search engine level. Each level is composed of elements of lower levels, i.e. a search engine is composed of documents and a document is composed of terms. The score of every engine is computed as described next. The final ranking of a resource is the sum of the weights of every term in the query, while the weight of a term with respect to a source is computed by the product of Inverse Resource Frequency (IRF) and Term Weight Frequency (TWF). IRF is defined as the inverse frequency of terms present in each resource and it is a generalization of the inverse document frequency, that measures the frequency of terms in retrieved documents. TWF is defined as the sum of all the values of previous weight values.

The last presented work of resources selection task is Bellogin et al. [2013], that has tested three different approaches:

1. Similarity between query's and results' categories. They used the Open Directory Project (ODP)[62] to get the categories associated to each resource and to every query. They then computed similarities between the two lists of categories using cosine [Tata and Patel, 2007] and Jaccard similarities [Hamers et al., 1989]. The source are ranked according to these similarities.

2. Retrieval model based: this strategy concatenates all the snippets from each resource and indexes them as a single document, so that when a query is issued, the aggregated documents (resources) are ranked according to their relevance with respect to the query.

3. Hybrid approach: it aggregates the two previous scores using a Borda voting mechanism [Dwork et al., 2001], where each document is given a number of votes inversely proportional to its ranking.

Every approach presented to the Federated Web Search resource selection track need to have access to the whole collection in order to compute the score of each retrieval source. These approaches are not applicable to this thesis because we would need to access every documents collection we are retrieving information from, and we can not access every possible result of each retrieval engine before sending the query to them.

We will consider a similar approach to the document-centric approach of Balog et al. [2012] with a difference: it uses the ranking of documents, while we will test a number of different scores (ranking, relevance, etc).

The sequential execution of several retrieval engines means that first a RE is requested, then other and so on until every RE has been requested. It is a very common strategy, and we will adopt this approach (see section 4.2.4) in this thesis as far as voice and image queries are concerned. Voice queries will be transcribed and then used as textual search as is proposed in the work of Hauptmann et al. [2002]. A similar approach to Lana-Serrano et al. [2011] will be adopted for image processing in this thesis. We will use two retrieval engines to extract tokens and objects (concepts) present in the image. This extraction results in a text, which will be used as textual query to request information from textual retrieval engines.

[62] http://www.dmoz.org/ accessed at 23/07/2015

There are some works [Balog et al., 2012; Chernov et al., 2006] that use the content of the query (terms) to analyze which *RE* to request. We consider to use not only its content but also another features such as structural features of textual queries: length, number of verbs, number of named entities appearing in the query, etc.

Although we will perform iterative retrieval with voice and image queries (see section 4.2.4), the approaches that divide the query according to their elements' modes are the most interesting [Demner-Fushman et al., 2012; Renaud and Azzopardi, 2012; Romberg et al., 2012]. We will consider the application of this approach. When we receive a combined query, it will be split into modes (text, image, video, audio) and then each mode is separately processed.

An example of sequential source selection strategy for a query combining text and image is shown in figure 2.7. The query is divided into two parts: the image, that is used for requesting the image retrieval engines, and the text, which is combined with part of the image REs results (text and concepts present in the image) for requesting the text REs. These REs return a set of results (documents). The image REs also returned the low-level features of the image. These features are used for requesting a CBIR engine, which also returns results (documents).

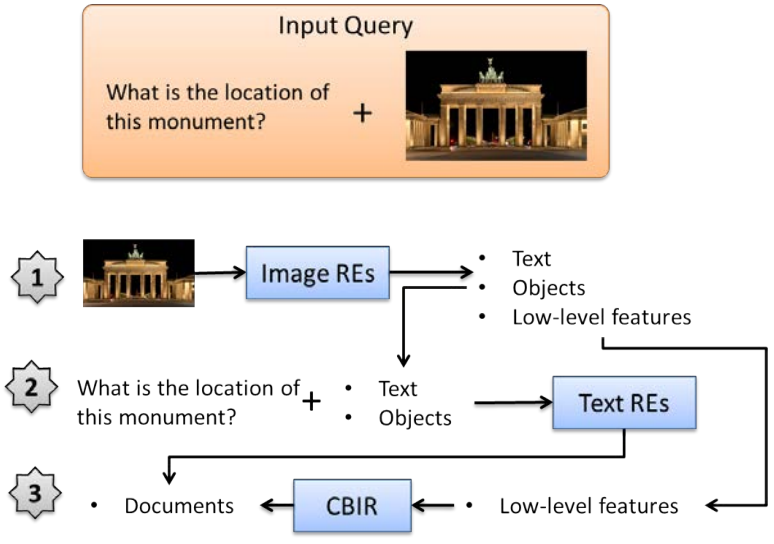


Figure 2.7: Example of sequential source selection strategy for a query combining text and image modes.

2.5 How are the results merged?

Whenever more than one RE is requested, there is a results' set coming from each one. Because there are more than a results' set, we must process them to get a single results' set which is returned to the user. This process usually consists of a combination or aggregation of different sets, which are used in two cases: when requesting several retrieval systems each of which returns a result set or when users want to obtain relevant information from several results (either from different sources or not). The second approach is studied in the field of *aggregated search*, where the main purpose is to offer users 'structures' of information from different sources or results to complete their information needs (or queries).

The aggregation approach is used in Bessai-Mechmache and Alimazighi [2012] to construct virtual elements containing relevant and non-redundant information. The fusion is performed by aggregating XML elements to make up a single result. A similar approach is used in Nottelmann and Fuhr [2003] that makes a XML-based fusion adding all the results in a final XML result.

A simpler approach is to organize the results randomly as in Chernov et al. [2006].

Results can also be organized based on their scores but this implies that scoring criteria must be unified to be homogeneous among different REs. The work of Arampatzis et al. [2011] fuses two scores: one from textual retrieval (based on TF-IDF weights) and one from visual retrieval ($DESC_{ij}$ explained in section 2.4). In Romberg et al. [2012] two retrieval engines are used (one for images and one for texts) that are combined using a linear combination.

Apart from these, there are more complex approaches which determine the new results order using machine learning techniques. Hong and Si [2012] uses a central index containing a summary of every document of the collections and three merging strategies: (i) Semisupervised Learning (SSL) merging [Si and Callan, 2003] that computes comparable scores of each document. It computes the final score using two scores: the score obtained from searching the document in the central index and the score of the original source. If there are no overlapping documents (an overlapping document is a document that has a score on both lists) between the results of central index and original source, then the score is computed using $S_{ij} = a_j \cdot R_{ij} + b_j$, where R_{ij} is the

source-specific ranking of document d_i in collection C_j and a_j, b_j are parameters depending on query and information source; (ii) Sample Agglomerate Fitting Estimate Merging (SAFE) [Shokouhi and Zobel, 2009] that solves the problem of SSL when there are few overlapping documents. It estimates the ranking of unoverlapping documents by making each overlapping document to represent N documents in the central index. Then the scores are computed in the same way as SSL; and (iii) Mixture of Retrieval Models (MORM) forwards the query to the selected information sources (using a selection algorithm) and to the centralized sample database. Each source will return a ranked list of documents and the centralized index retrieves a set of ranked lists of sample documents using predetermined algorithms. MoRM tries to learn a mapping between source-specific document ranks and the centralized document scores. All comparable scores of a document are combined using a set of combination weights learned from a training dataset. These final scores are used to rank documents.

There are other systems that fuse the results at the visualization step. They retrieve documents and display them altogether in a single results' list or set. Works using this approach are those that retrieve information from web search engines (and their verticals) such as Arguello et al. [2012] that does not provide specific information about its fusion technique, but the result is a list combining every mode. Two works that do not perform a direct fusion after retrieval are Nottelmann and Fuhr [2003] and Steiner et al. [2012]. Both works use a specific language for representing documents. Therefore, when the matching between query and documents is done, there is only one results' set (list). The fusion is made when the documents are mapped to the language format.

Finally works implementing new techniques for fusion of results are highlighted. Romberg et al. [2012] uses a co-occurrences matrix. Each row of the matrix represents an image and each column represents an image low-level feature. Therefore, each position of the matrix contains the value of a concrete feature of that image. The AENN algorithm is defined in Balog et al. [2012]. It uses a *central broker* that besides assigning a score to each source it also defines the number of results that are retrieved from each source. This algorithm is based on a linear combination of two language models to rank sources: a collection centric approach (CC) and a document centric approach (DC) (explained in section 2.4).

In this division we can include all those systems that perform retrieval through joint indexes, being the fusion in these cases a prior fusion performed before the retrieval

process by representing the documents in the index feature vector space: Yang et al. [2002], embracing multimodal documents and links between them in an index known as Multifaceted Knowledge Base (MKB), Demner-Fushman et al. [2012], combining text and visual features in a multimodal index, or Marchand-Maillet et al. [2011], defining a unified model of matrix where it aggregates documents, concepts and users.

The objective of the second task of the aforementioned forum (Federated Web Search Track) is to rearrange and combine the results obtained from every search engine for each query. Only to give an overview of the best systems participating in this task they are described next.

In Guan et al. [2013], the score of each document is computed as a linear combination of similarities between the query and different fields in a combined index. The score of every field of the index is computed using the Okapi BM25 retrieval algorithm [Jones et al., 2000].

In Pal and Mitra [2013] they also used a linear combination to reorder the results. The two combined factors are the original ranking obtained from the search engine and the search engine score value obtained according to the source selection strategy (see section 2.4).

Another work is Mourao and Magalhaes [2013], that claims that *'its idea is based on the known pressure for Web search engines to put the most relevant documents at the very top of their ranks and the intuition that relevance of a document should increase as it appears on more search engines'*. This work considers that each list of results from an engine has a score that is equal to the ranking. After, it looks for results that appear in more than one list to add the scores of every list. They are using three fusion methods: rank-based fusion function (RFF) (score of a document is the sum of inverse of its ranking in every RE), Condorcet Fuse [Montague and Aslam, 2002] and their proposed method: Inverse Square Rank (ISR) fusion algorithm (the score of a document is the sum of the inverse square of its ranking).

The work of Bota et al. [2014] presents a framework for developing *composite retrieval*. This type of retrieval is based on the request of heterogeneous web search systems and the generation of a combined response, which is composed of a set of bundles each one encompassing results from different verticals. The result should contain a set of bundles that minimize a utility function based on four criteria: relevance, topical cohesion, topical diversity and vertical diversity.

The most interesting work concerning results' fusion is the approach presented in Wu and Crestani [2015]. This work describes a geometric space where the documents are associated to the query and the RE, so the geometric space is a hypercube of dimension n (number of documents). Each RE returns a vector with a value associated for each result, being zero when this result is not returned by the RE. Once these vectors are returned, two fusion approaches are studied. The first approach is based on centroids, which calculates the centroids of the resulting vectors of each RE. These centroids are the returned results. The second approach is based on a linear combination, which combines the vectors returned from each RE. The problem of this approach is the assignment of the weights: if there is no previous information, weights must be randomly assigned.

This section has introduced interesting works about results' fusion. Important are the works that combine results based on scores [Arampatzis et al., 2011; Balog et al., 2012; Romberg et al., 2012]. Regarding more complex approaches, Romberg et al. [2012] uses a table of co-occurrences where it represents visual words (columns) that are present in each media item (rows) and systems using joint indexes, being the fusion in these cases a prior fusion performed before the retrieval process by representing the documents in the index feature vector space [Demner-Fushman et al., 2012; Marchand-Maillet et al., 2011; Yang et al., 2002].

We will apply the same approach as Pal and Mitra [2013] for combining the results with a difference: the ranking of our sources is determined by the order in the rules of our handling strategy. We will also base our approach on Guan et al. [2013], which defines a linear combination where the weights of every engine will be changed in our case by the order of engines in the handler rules.

A priori, we will implement a simple algorithm to focus on the selection of sources. Therefore Round Robin algorithm [Silberschatz et al., 2008] is used in a first step. Future enhancements will consider incorporating some of the techniques presented here.

2.6 Semantic Knowledge

Currently multimedia information retrieval is done in many cases using semantic information. It is similar to metadata based retrieval, where metadata are obtained from semantic knowledge bases such as ontologies or taxonomies. Due to this and because

one of the components of the formal model is based on ontologies, this section is devoted to describe semantic-based IR approaches. Currently, there is an increasing availability of semantically annotated multimedia resources mainly due to the improvements of automatic semantic annotation systems as well as annotations generated by social media users. The annotations generated by users through social networks can be used as a source of semantic information.

As explained in the presentation of Ivan Cantador[63], semantic knowledge bases have evolved: first they were *Bags of words* containing uncategorized terms; then they converted into *Taxonomies* by adding categories and hierarchical relations; the next evolution are the *Thesauri*, which besides categories and fixed hierarchical relations, it also includes associative relations; the last improvement gets the *Ontologies*, which encompasses classes, instances, arbitrary semantic relations and rules for performing inference.

According to Pino and Di Salvo [2011], the approaches performing semantic multimedia retrieval are classified according to the following aspects:

- *Multimedia retrieval based on annotations, relevance feedback and concepts*: it is similar to metadata-based search. The documents are retrieved based on the similarity of their annotations and the annotations of the query. It is mainly focused on similarity of concepts.

- *Retrieval based on multimedia ontologies*: it is similar to the previous type, but the multimedia objects are annotated with semantic information taken from an ontology.

- *Recognizing semantic framework for intrinsic objects*: it refers to the automatic annotation of multimedia objects in order to use these annotations for retrieval. (Semi)automatic systems which make semantic annotation are encompassed in this type.

- *Combination of multimedia ontologies using domain specific ontologies*: it refers to the combination of domain-specific ontologies. Each ontology offers certain

[63]https://canal.uned.es/mmobj/index/id/24093/hash/cc42acc8ce334185e0193753adb6cb77 accessed at 23/07/2015

information which complements the other ontologies. The whole combination creates a global semantic environment to be integrated in the system.

We consider the first two cases are very similar, just that in the second case the types of annotations appear in an ontology rather than in other sites.

2.6.1 Annotation-based and ontology-based retrieval

The Mediamill system [Worring et al., 2007] is a semantic video search engine. Mediamill is meant for exploratory search which starts with a textual query. It is used to retrieve a set of video shots (keyframes). These keyframes are displayed in a specific view, known as RotorBrowser that displays images like the blades of a windmill where there are 4 blades: (i) textual blade, which contains similar keyframes based on their annotations; (ii) low-level features blade containing similar images according to their low level features; (iii) timeline blade, showing chronological-ordered keyframes; and (iv) semantic blade, that contains images which semantic content is similar.

Another work is Shah et al. [2002], in which a framework called OWLIR is described. This framework is an approach to the retrieval of text documents that are accompanied by semantic annotations. The annotations have DAML&OIL format. Each DAML&OIL tag is used as an indexing term in order to allow retrieval of documents based on their semantic markup information. The framework is able to extract and exploit semantic information using an event ontology and AeroText[64], a commercial suit of text mining applications.

The ESCRIRE project and EsCosServer architecture [Medina-Ramírez, 2007] explore domain ontologies, semantic annotations and semantic descriptions of resources to improve information retrieval. To do so, it processes documents, annotates them semantically through a domain ontology and stores this knowledge into the EsCorServer. EsCorServer architecture allows the use of heterogeneous sources of information to be represented, handled, queried and diffused. It converts the ESCRIRE ontology into RDFS (resource description framework schema) for being usable. The queries are natural language text and the system returns text documents annotated with additional information of the ontology in an aggregate way.

[64]http://www.rocketsoftware.com/solutions/enterprise-search-and-text-analytics accessed at 23/07/2015

Castells et al. [2007] defines a semantic search algorithm based on a modified vector-space approach and using keyword-based queries. These keywords are gathered by a linear combination of results, which are combined using the COMBSUM algorithm [Shaw and Fox, 1994]. A document score generated by this algorithm is the sum of every score of this document (in each of the different sources). It uses text queries (natural language, RDF, etc.) and returns full documents (text), which are related to concepts that have been previously obtained from the ontology.

PowerAqua is a semantic Q&A system [Lopez et al., 2012] in which the search for an answer is performed by searching for information distributed across semantic sources (ontologies and web accessible and free semantic systems). It accepts natural language queries, which are processed to obtain their linguistic information and to represent it as triplets to formalize the inter-dependencies between terms. The triplets are sent to a set of storage platforms (semantic sources) such as Watson SW Gateway[65], Virtuoso[66] and Sesame[67]. The ontological facts (triplets) are merged by means of three analysis: redundancy, intersection and union. Once the facts are merged, a set of ranking criteria to sort the list of results is applied. First the results are ranked based on the confidence of the mapping algorithm, i.e. if the mapping has been extracted using the original query term or by means of any of its synonyms or hypernyms; and how well the triplet from which the answer is extracted covers the information specified in the query. Then, answers obtained by means of the most popular semantic meaning (the one appearing in a higher number of ontologies) are ranked first. This system is limited to retrieve information from the ontologies because it gets the similarity calculated by triplets taken from query and semantic web (SW). No other documents than those included within the ontology are returned. Therefore, its main limitation is that each information that should be retrieved must be added to the ontology. This can elicit scalability problems.

2.6.2 Automatic semantic annotation approaches

The third aspect (*recognizing semantic framework for intrinsic objects*) mentioned by Pino and Di Salvo [2011] focuses on automatic annotation of resources. In Bracamonte

[65]Watson: http://watson.kmi.open.ac.uk/WatsonWUI/ accessed at 23/07/2015
[66]Virtuoso: http://virtuoso.openlinksw.com/ accessed at 23/07/2015
[67]Sesame: http://rdf4j.org/ accessed at 23/07/2015

[2013] an algorithm to automatically compute semantic annotations of multimedia elements is defined. It also allows filtering and grouping of the generated annotations. This proposal focuses on the design of scalable and effective solutions to enhance the description of multimedia objects on the web.

2.6.3 Ontology combination approaches

The fourth aspect (*combination of multimedia ontologies using domain specific ontologies*) is discussed. García and Celma [2005] makes a combination of ontologies by mapping OWL to MPEG7 and viceversa to achieve the integration of several ontologies and to make them accessible with the same criteria. It has been tested with three music domain ontologies using RDF query language (RDQL).

Linked Open Data (LOD)[68] is an iniciative, whose goal is to integrate the available open online semantic knowledge into a single unified semantic resource. The tutorial *How to Publish Linked Data on the Web*[69], by Chris Bizer, Richard Cyganiak and Tom Heath, offers a definitive introduction to create and publish Linked Data. LOD attempts to integrate the majority of publicly available semantic knowledge. It is not limited by closed-domain or homogeneous scenarios. LOD contained 1014 linked datasets (semantic knowledge bases such as ontologies) in April 2014 from different domains: Government, Publications, Life sciences, User-generated content, etc (extracted from the State of the LOD Cloud 2014[70]).

2.6.4 Semantic MIR systems comparison

The semantic approach we define in this thesis will cover two of the aforementioned aspects: the combination of multimedia ontologies using domain specific ontologies and retrieval based on multimedia ontologies. Our ontology will be composed by 30 smaller ontologies in order to define a whole multimedia and sports ontology (deeply explained in section 4.2.2). The ontology-based retrieval of our approach is different from those explained in related work. Our system will not match concepts in the query with concepts in the documents, but it will retrieve concepts from the ontology based on the query's content in addition to the documents that contain or refer to these concepts.

[68] http://linkeddata.org/ accessed at 23/07/2015

[69] http://linkeddata.org/docs/how-to-publish accessed at 23/07/2015

[70] `http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/` accessed at 23/07/2015.

2.7 User Behavior

Adaptive information retrieval has gained research interest because current systems can not be so rigid as to work the same way for all people at all times. Currently, information retrieval systems are much more frequently used than before and by many more people. Therefore, information needs are wider and systems must adapt to this variability.

Adaptive systems register user actions within the systems to improve retrieval accuracy. Each action is considered as an interaction. First of all, the user model based approaches, which process user interactions and generate user models are described. Although user modeling is out of the scope of this thesis (because we do not want to particularize the functionality for specific users), this type of systems are explained because they are interesting considering the techniques and the interactions they use. The rest of the works are characterized by using approaches considering interactions and can be classified into two types: (1) direct interactions-based approaches, which register information related to the searches (queries) or actions performed over the results (relevance judgments, results visualization, results filtering, etc.); and (2) indirect user actions, which refer to actions that users are not conscious of, such as eye tracking or lip and gesture motion.

2.7.1 User model based approaches

There are some works that create or adapt user models in order to classify users' behavior. Modeling user behavior or creating user models helps improving IR results by adapting system performance to a certain type of user. Its main disadvantage is that these models are limited to the type of users that are considered by the models. If they are too specific, they generate too many models. On the contrary, if they are too general, they obtain too few models.

In Mianowska and Nguyen [2011] a customization system to create user profiles that are adapted from the results of IR is presented. A user profile is defined by the interest of a user in terms that make up the profile. These terms are extracted from the results of IR. The main point of this work is that the models consider the time variable by decreasing the interest of a user when a term is not present in the latest results. There

is a clear disadvantage observed in this work: it does not change the functionality of the IR according to the profiles, but modifies user profiles based on the results of IR.

In Rekha et al. [2011] two approaches are studied: the generation of a user model by means of clicks and the adaptation of ranking functions. The first part is accomplished by means of analyzing three features: query expansion, dwell time (it is defined as *'The period of time that a system or element of a system remains in a given state'*[71], in this case it refers to the time a user spends visualizing a result or a results list) and copy/cut operations (from the results' text). Using these features and considering the information retrieval as a decision optimization problem, they propose a formal decision theoretic framework, which focuses on minimizing the loss function (Bayesian Risk [Hannan, 1957]) by every action previously done by users and every response from the system. This minimization modifies the functionality of the system.

The User-Centered Adaptive Information Retrieval (UCAIR) system is presented in Shen et al. [2005]. This system uses a decision theoretic framework to perform implicit user modeling. The model is generated using the search history log (where the queries are stored). For each user action, the system accomplishes two actions: updating the user model and finding the best response that minimizes a loss function.

2.7.2 Approaches using query log information

The registration of queries is another common logged information in interactive systems. Query log information refers to the queries sent by users and the associated information about them such as geolocalization, timestamp when the interaction was created, device used to generate it, etc. The query is normally stored by its content, but it is also possible to store other information, such as entities (person, location, organization) or concepts contained in the query, etc. The way it is stored also distinguishes it.

In Shen and Zhai [2003] historical queries, also known as previously performed queries, help the retrieval by two techniques: the first technique returns a combination of the results of every historical (previously performed) query getting a final results' list, while the second technique generates a joint query model of every previously performed query and uses it for requesting the retrieval system. Historical queries are also registered in Arguello et al. [2012] by means of the Lemur Query-Log toolkit. In Golovchinsky and Diriye [2011] search query history is logged and displayed in order

[71] Taken from Wiktionary (http://en.wiktionary.org/wiki/dwell_time) at 23/07/2015.

to handle collaborative search. This type of search is defined as multiple users completing a search task through the system. Therefore, it is interesting to access queries previously performed by other users as well as their obtained results' lists.

Nowadays log files analysis is becoming popular in interactive systems. There is a lot of research done in this area such as Ahn et al. [2011] that analyzes log file content to extract user behavior patterns. In this work the patterns are extracted to create user models in order to use them together with two adaptive exploratory search systems (ESS): TaskSieve [Ahn et al., 2008] and VIBE [Ahn and Btusilovsky, 2009]. The patterns are obtained using the following interactions: login and logout, search, overview, find subset, examine document and points of interest[72] (POI) activities (only with VIBE). These interactions are used for extracting patterns by two approaches: overview of temporal sequences and alignment of specific sequences of interactions.

2.7.3 Interaction-based approaches

Interactions are actions that the user performs inside or during the use of information search systems. Using this information to adapt system functionality to the user is almost mandatory. User actions indicate its intention to retrieve information [de Vries, 1998]. The searches made previously are not taken into consideration because these works have been presented in the previous section. Therefore, this section focuses on the interactions associated with search results.

Just as there are a lot of different types of interactions (relevance feedback, dwell time, cursor movements, gesture analysis, and others), we must also distinguish the information associated with each interaction. It is not the same knowing that the user viewed a result which was second on the list than knowing the type of the result, its URL, its relevant terms, etc. Because of this, this section will become not only a presentation of the different interactions that the systems register, but also the information associated with them and how the interactions and their information is processed or used by the systems.

The interactions that every system records are different and depend on its purpose. Relevance judgments (relevance feedback) are the most used user interactions as an indicator of user behavior and needs. These interactions are widely used to identify or classify user behavior. Mandl and Womser-Hacker [2003] ensures that *'user relevance*

[72] A point of interest (POI) is a part of the interface that focuses the interest of users.

feedback can be considered to be the best technique to improve the results of information retrieval systems'. These interactions can be recorded directly asking users to mark documents or results that seem relevant or indirectly through other interactions, such as considering that each result is relevant in function of the time that is spent in each document.

As an example, the work of Kong et al. [2013] analyzed the relevance of a document (or section) based on four categories of user behavior features: dwell time that users spend viewing a result (cumulative dwell time, averaged dwell time), clicks that users make on documents or sections (rollover and see link probabilities together with their deviation), text highlighting while viewing a result and text copying while viewing a result. The complete list of user features is shown in [Kong et al., 2013, Fig. 3.2]. Two ranking algorithms, RandomForests [Breiman, 2001] and RankBoost [Freund et al., 2003], are used to determine which sections (or documents) are more relevant. The input of each algorithm are the sections and the output are their rankings.

The MIMOR system is described in Womser-Hacker [1996]. The main advantage of MIMOR is that it combines several information retrieval systems, where the influence of each system is based upon its previous performance for the user measured by the relevance feedback. The basic idea of the MIMOR model is to optimize its quality through learning from user feedback. Since it uses several retrieval engines, a document obtains a score (retrieval status value - RSV) from each retrieval engine, which are combined using a set of weights. Finally, the document's score is computed by dividing the sum by the number of retrieval engines (average). The weight of a document-RE pair is modified using two parameters: a learning rate (with values between $\{0, 1\}$) and the user relevance feedback assigned to the document (with value $\in [-1, 1]$).

Interesting is Mandl and Womser-Hacker [2003] that adapts MIMOR by creating new models to avoid initialization issues. It considers that a user model is not good enough when there are not enough interactions. This leads to the creation of a general model (considering interactions from all users) that is used whenever a user has less interactions than a threshold. The weight of a document used in Womser-Hacker [1996] is converted in a linear combination between the public (general) model weight and the particular (user dependent) model weight: $w_i = (p_i \cdot w_{private,i} + (1 - p_i) \cdot w_{public,i}$, where $w_{private,i}$ and $w_{public,i}$ are the private and public model weights and $p(j)$ is a variable depending on the interactions made over the document.

Another work using relevance feedback is Hoi and Wu [2011], that refines the retrieval results from user's relevance feedback. A semi-supervised active learning technique (taken from Hoi et al. [2009]) used relevance information to identify image examples (for labeling), and employed the semi-supervised SVM to re-rank the retrieval results based on image examples.

Yang et al. [2002] registered relevance feedback (positive and negative values) in order to filter the results offered to the user. It is worth remarking that only relevance feedback of a user was used for filtering results of this user. The results were filtered by computing new scores using two self-defined algorithms: 'discover' and 'distill' algorithms, which were applied to the positive and negative documents (using them as seeds) marked by the user. Then, the final score of a document was computed by subtracting the negative score (obtained from the application of the algorithm to the negative seeds) to the positive score (positive seeds algorithms' results).

On the other hand, indirect relevance by documents visits and click analysis was performed in Buccio et al. [2010]. This approach gathered user interactions on the n first retrieval results and computed a matrix extracting k features of user behavior. It used the Principal Component Analysis [Jolliffe, 1986] for extracting these features. At the same time, it also computed a vector of features for each of the first m retrieved results. Then, the m first results were re-ranked using the matrix and the vector of each result (matrix product). The values of n, m and k were computed experimentally (it uses $n = 3$, $m = 10$ and $k = not\ specified$).

In relation to other interactions (different from relevance feedback) the most common are *analysis of clicks*. In this sense, the work of Agichtein et al. [2006] adapted a simple approach of ignoring the original rankers' scores, and instead simply merged the rank orders. The final ranking of the results was obtained from the combination of an implicit ranking obtained from the characteristics of user interactions and the original document ranking. User behavior features were associated with each query-result pair (result was determined by the result's url) and were divided into three types: (i) click-through, features associated with each click event such as probability or frequency of clicking a result; (ii) browsing, features associated to dwell time and followed links; and (iii) query-text, features associated with similarities between result and query. These classified feature vectors were used as input to the RankNet [Burges et al., 2005] train-

ing algorithm which produced a trained user behavior model. This model was used to improve the retrieval results by generating an implicit feedback value for each result.

In Huang [2011] there are interactions (called 'page-level interactions') that are logged on the client side and are: cursor activity, parallel browsing behavior, web browser meta-data and dwell time. A complete list of these interactions can be found in [Huang, 2011, Pag. 3]. Dwell time in the results and content pages is registered in Liu et al. [2011] and Beckers and Fuhr [2010]. Not only dwell time in the results but also scroll-down and scroll-up movements are logged in Buccio et al. [2010].

Something that has gained interest in last years is eye tracking, i.e monitoring where the user is looking at different moments. These interactions have the advantage that there is no direct user intervention needed while they are logged automatically. Users always look at something in the interface and they are not distracted from their task to make some specific actions. A work that performs eye tracking is Cole et al. [2011]. It determines which terms the user is looking at and analyzes them. With this information it performs sequences of fixation (eye movement analysis technique). In Kules et al. [2009] eye tracking analyzes what do users use in a faceted search system making exploratory search. User-oriented eye-tracking-based evaluation of an interactive search system is presented in Beckers and Fuhr [2010]. Both works split the screen into eight areas of interest (AoI) to determine which are the more viewed areas and to analyze where the users are fixing their attention.

Steichen et al. [2013] is interesting because it uses eye tracking to design new display systems adapted to the user. They use two basic display techniques to evaluate their proposal, while for the analysis of interactions they use classification techniques (machine learning) using the tool WEKA[73]. Another work that uses these interactions, along with clicks, is Joachims et al. [2005]. While clicks are used to determine the relevance of the results, eye tracking analysis determines the cognitive processes and the information needs of users. Using both strategies it makes a thorough analysis of user behavior during the review of results, the relationship between clicks and relevance of the results and analyzes the relationship between clicks and zones looked at by the user.

Finally it is interesting to present the work of Santos Jr and Nguyen [2009]. This work adapts the system functionality based on user interests, context and preferences

[73]http://www.cs.waikato.ac.nz/ml/weka/ accessed at 23/07/2015

instead of using only their interactions. Interests are determined in a set composed by concepts and levels of interest. The concepts found in a set of interests are obtained from documents that the user has identified as relevant. The user interest in each of these concepts is determined by the number of retrieved relevant results and the number of retrieved documents containing it. The context is represented as a directed acyclic graph (DAG) [Kalisch and Bühlmann, 2007] similarly to a document graph. The graph contains two kinds of nodes: concepts nodes and relations nodes. The context graph is constructed by finding the intersection of all retrieved relevant documents' graphs. The preferences are defined as a Bayesian network [Ben-Gal, 2007]. This network defines the way a user wants to form a query. This preferences are used for modifying the query, which adapts the retrieval process. The input query of the system is modified based on user interest, context and preferences and the new query is matched against the documents based on the number of concepts and relation nodes present in query and document graphs and the total number of concepts and relation nodes of the query.

Besides, video recording of user sessions is another indirect interaction. It is interesting because it is possible to perform a postprocessing of the user behavior by analyzing the recorded images. In Malla et al. [2011] user sessions are recorded to analyze user behavior during task completion.

Interactive systems are characterized by the participation of users so the registration (as in Renaud and Azzopardi [2012]) or a previous training (as in Sushmita [2012] or Malla et al. [2011]) are interesting things to consider when reviewing interactive systems. Last but not least some works consider interviews with users such as Kules et al. [2009] and Sushmita [2012] or surveys to obtain users perception such as Arguello et al. [2012], which uses a user experience and exit questionnaires, or Kules et al. [2009], which uses surveys to determine user perception about the exploratory search.

Reviewing works fulfilling interactive evaluations, it is interesting to mention the work of Renaud and Azzopardi [2012], that introduces the SCAMP (Search ConfigurAtor for experiMenting with PuppyIR) system. Its goal is to generate interactive IR experiments registering user interactions. It allows to log the next interactions: registration, consent, survey, tracking of tasks and participants and questionnaires (post or preexperiments).

The adaptation of systems from user behavior encompasses many different areas. To the best of our knowledge, multimodal IR based on user interactions has been

addressed but we can not apply these approaches directly. Some works covered user models [Rekha et al., 2011] while others treated engine selection and results' fusion [Womser-Hacker, 1996]. However, there is no work that covers every aspect. This thesis tries to fill the gap of adapting the IR functionality based on user interactions without particularizing this adaptation by means of user models.

Some works do not modify the IR functionality: Joachims et al. [2005] aim to determine to what extent the relevance of a result can be determined and in Steichen et al. [2013] they are modifying the display modes, although the classification techniques are taken into account.

The main limitation of Kong et al. [2013] is that it does not adapt the IR functionality itself, but it just reranks the final results. It could be seen as an adaptation, but there is no management of different sources. Mandl and Womser-Hacker [2003] presents a severe disadvantage because it needs to know the similarity between all the retrieved documents with each retrieval engine, and this information is not available for us. We would consider something similar using different values for document-RE pairs. Our work differs from Santos Jr and Nguyen [2009] because we do not have access to modify the IR system to optimize the query format. We consider that IR systems are black boxes and we tailor the query to each system.

Among the techniques presented some are interesting and will be applied in our work. These techniques are:

- The idea of generating a model using the search history and using it to modify the REs order and the ranking of the results is taken from Shen et al. [2005].

- The approach of Womser-Hacker [1996] (MIMOR system) optimizes its quality through learning from user feedback. The influence of each system is based upon its previous performance for the user measured by the relevance feedback. From Womser-Hacker [1996] we adopt relevance feedback as elements for measuring the performance of a RE, but the scores assigned to each source are different.

- The document-centric approach of Balog [2013] is applied with a modification. It uses the score of documents, while we used a number of different scores (explained in section 6.4.2).

The number of interactions that are used in this thesis has been limited for two reasons. First, the GUI determines the interactions that can be registered. Depending on how it will be implemented, it is possible to record interactions on the client side or not. On the other hand, we have decided to keep low the number of interactions to check the developments to be made. Thus, if the results are good, we can generalize to a larger number of interactions. We will focus on query history as well as direct relevance feedback and results' interactions. This thesis considers interesting that user should register in the system, although using it anonymously will be also possible. User registration is interesting for identifying users and allowing future user modeling. Final survey is also interesting because it is a clear indicator of user's impression.

By contrast, this thesis does not consider indirect relevance based on dwell time because it is not clear that displaying a result qualifies it as relevant. It may be that the user has accessed it to see if it was relevant and it is not. The same consideration is applied for scroll up, scroll down, cursor activity. We can not assure that scrolling a result down (until bottom) or by moving the cursor over the content of a result make this result relevant. Therefore, these interactions are not considered in our approach. Other client-side interactions, such as eye tracking, gestures, lip motion, speech or facial expression are not considered in this thesis because the analysis of these interactions is out of the scope of it.

2.8 Federated Web Search Track (2012 and 2013)

The Federated Web Search (FedWeb2013) Track has been previously mentioned in section 2.1.1 where the collection that was used in the track has been described and in sections 2.4 and 2.5 where some characteristics of the works presented to the track were highlighted.

The handler and results' fusion modules (see section 1.3) are present when querying more than one RE. Since we will focus on adapting these two modules, a brief description and summary of the FedWeb2013, which involved federated search systems, is given.

Comparative evaluation (benchmarking) is important to know to what extent our system meets the current state of the research area. Evaluation forums offer an unbeatable framework for this simple and unified benchmarking.

There are not many evaluation forums that cover multimodal information retrieval. Some forums are: Interactive track of Cross-Language Evaluation Forum (iCLEF), Interactive Track of Text Retrieval Evaluation Conference (TREC) or TREC Web Track (from 2009 to 2014). The works participating in these forums are not described because:

- iCLEF: cross-language search capabilities were studied from a user-inclusive perspective, but the works did not consider multiple retrieval engines (neither the strategies of sources selection nor results merging).
- Interactive Track 2005 (TREC): it had the same problem as iCLEF. The systems do not use multiple retrieval engines, so the handler and fusion modules were not considered.
- Web Track 2013 (TREC): it used several collections. Last collection they used was CluwWeb12 Dataset[74]. The main problem is that the collection does not have a semantic knowledge base that related the documents.

Federated Web Search Track forum is the nearest to the development this thesis. It focuses on federated search systems, which covers information retrieval from several retrieval engines. Furthermore, federated search systems have an important similarity with our system: several different and heterogeneous retrieval systems are used.

A complete overview of the Federated Web Search track can be found in Demeester et al. [2013]. It points out that *'the generation of big scale search engines depends more and more on the combination of multiple sources. A web search engine can combine results of several verticals such as: videos, books, images, scientific articles, shopping, logs, news, music, maps, ads, Q&A, job offers, social networks, etc.*

The use of many systems yields a new problem that must be taken into account:

'In general, the search results of each sources differ in the offered fragments, the provided additional information and the ranking approach used.'

In this sense, it helps to address an existing problem:

'Federated search allows the inclusion of hidden web collection results that are not easily accesible by other ways.'

The track was composed by two tasks:

[74] `http://boston.lti.cs.cmu.edu/clueweb12/` accessed at 23/07/2015

- Source selection: *'its goal is to select the right resources from a large number of independent search engines given a query. [...] (157 search engines)'* [Demeester et al., 2013]. In the resources selection task participants have to classify 157 engines for each topic without having access to the corresponding results. The participants must extract a set of retrieval engines. We describe in section 2.8 the best systems participating in the task that submitted any run to the source selection task.

- Results merging: *'its goal is to combine the results of several search engines into a single ranked list. [...] The result pages include titles, snippet summaries, hyperlinks, and possibly thumbnail images, all of which were used by participants for reranking and merging'* [Demeester et al., 2013]. The objective of this task is to rearrange and combine the results obtained from every search engine for each query. The training collection offered results for each training topic, so that the systems could train their approaches. Only to give an overview of the best systems participating in this task they are described in section 2.8.

The TREC Federated Web Search (FedWeb) Track 2013 offers a collection that provides a standardized framework allowing researches to compare different approaches easily. It is different from artificially created collections because it provides results obtained from 157 real web search engines divided into 24 categories (ranging from news, academic articles and images to jokes and lyrics). The collection contains both the search result snippets $(1,973,591)$ and the pages $(1,894,463)$ the search results link to (the HTML of the corresponding web pages).

The categories (completely listed in [Demeester et al., 2013, Table 3]) range from *Academic* or *News* to more unusual categories such as *Games*, *Software* or *Encyclopedia*. A complete list of search engines can be found in [Demeester et al., 2013, Appendix A]. The engines are varied in domain and category such as: *arXiv.org*, *Wikipedia*, *SourceForge* or *Amazon*.

2.9 Discussion

This chapter has introduced many research works in different research areas. Therefore, we will proceed with a summary highlighting the main features learned. The most

outstanding works are presented in table 2.5, which summarizes these works by means of five characteristics: accepted modes in the query, modes of handled documents, source selection strategy, results' combination strategy and interactivity of the system.

The table consists of six columns. The first column shows the name of the system (if it has a specific name) and the reference of the work where the system is described. The second column describes the different modes in the query that each system accepts. The possible values for this field include not only accepted modes themselves (Text, Image, Video, Audio) but also contains other values as RUCoD and DAML&OIL referring to specific languages for defining queries. The third column contains information on two things: the modes of documents that can handle the system and the way in which documents are handled. The fourth column provides information about the strategy (if any) used for the selection of sources. If the work does not provide source selection strategy it is assigned a null value represented by two dashes ('–'). The fifth column describes results' combination strategies only in systems requesting several different sources. The use of null value ('–') is also contemplated if the system does not use multiple sources or the results are not combined. The last column shows the interactivity (*Interac.*) of the system. This column shows only a *YES* value in works if they consider user interactions and null value ('–') in those which do not. The different interactions that each system records and processes are omitted to simplify the table.

System	Query Modes	Multimodal Information	Handler	Fusion	Interac.
Bessai-Mechmache and Alimazighi [2012]	Text	Text	—	YES	—
Sushmita [2012]	Text	Web, Image, News, and Wiki	—	YES	YES
Arguello et al. [2012]	Text	Text, Images & Videos	—	YES	YES
Beckers and Fuhr [2010]	Text	Text & Image	—	—	YES
JIGSAW [Görg et al., 2010]	Text	Text	—	—	YES
Hu et al. [2011]	Text	Music	—	YES	—
Malla et al. [2011]	Image & Text	Web (Bing)	—	YES	YES

Nottelmann and Fuhr [2003]	DAML & OIL (D&O)	Multimedia D&O Represented	YES	YES	—
Hong and Si [2012]	Text	Text	YES	YES	—
SIRE [Hoi and Wu, 2011]	Text & Image	Image & Text	—	—	YES
Octopus [Yang et al., 2002]	Text, Image & Video	Combined Index	—	—	YES
SCAMP [Renaud and Azzopardi, 2012]	Text	Web Results	—	YES	YES
Steiner et al. [2012]	RUCoD	Multimedia Elements (RUCoD)	YES	YES	—
Romberg et al. [2012]	Image	Image	—	YES	—
Yilmaz et al. [2012]	Text	Videos	—	YES	—
de Vries [1998]	Text and Music	Joint feature database	—	YES	YES
Thesis development	Multimodal	Multimodal[75]	Rule-based	Simple Approach	YES

Table 2.5: Summary of the most relevant works

Regarding the review performed in this section, there are many works that make multimodal information retrieval, although most of them only work with two different modes. Furthermore, in many cases one of the modes is only textual information associated to multimedia elements. On the other hand, there are articles that review interactions between users and system and that study the modification of system functionality suited to them. These works have the drawback that most of them use monomodal collections.

Next, the techniques that will be adopted in this thesis are enumerated.

[75]Since the retrieval systems are externally created, the management of multimodal information is also declined to them.

- Regarding multimodal collections (more than two modes), we found only collection of Federated Web Search Track (see section 2.8), which has the limitation that no semantic information (linking documents) is available.

- The collection management is declined to the external REs. Retrieval techniques are not studied directly in this thesis and they are black boxes for us.

- It would be interesting not to limit the number of modes in a query. For this reason, we are going to borrow the approaches of Yang et al. [2002] and Marchand-Maillet et al. [2011]. We will focus on those approaches to define and design the management of multimodal query.

- The engine selection (handler) strategy will be based on a mixture of the approach of Demner-Fushman et al. [2012]; Renaud and Azzopardi [2012] that use the content of the query (terms) to analyze which RE to request and the a sequential execution approach similar to the approach used in Hauptmann et al. [2002].

- The main advantage of Octopus [Yang et al., 2002] is that it represents multimodal information in a single index with low-level features: Multifaceted Knowledge Base (MKB). It is an advantage because it models and considers not only content of the documents but also relations between them. Semantic relations can be considered inside them.

- The idea of generating a model using the search history and using it to modify the REs order and the ranking of the results is taken from Shen et al. [2005].

- The approach of Womser-Hacker [1996] (MIMOR system) optimizes its quality through learning from user feedback. The influence of each system is based upon its previous performance for the user measured by the relevance feedback. From Womser-Hacker [1996] we adopt relevance feedback as elements for measuring the performance of a RE, but the scores assigned to each source are different.

- The document-centric approach of Balog [2013] is applied with a modification. It uses the score of documents, while we used a number of different scores (explained in section 6.4.2).

To the best of our knowledge there is no work that considers multimodal information retrieval (texts, images, videos and audios) semantically related while taking into account the user and their behavior within the system. The existence of this gap justifies the developments of this thesis.

3

A Model to describe MIR systems

One of the objectives of this thesis is to define a formal model that allows designers to configure different IR frameworks considering the revised elements (chapter 2). The definition starts with an abstract architecture of a IMIR system (see figure 3.2), later moving to particularize formally every component: multimodal information, query, retrieval engines, handler, results' fusion, semantic knowledge and interactivity.

3.1 Environment

A formal model is necessary because we need a way of describing a given reality accurately, without any ambiguity. In our case, the objective is to generate different IMIR systems using the model described in this chapter. By using such a model, a standardization among all systems that are based on it is achieved. This standardization allows comparison of different systems and the ability to exchange modules between different systems as well as to isolate components that could be easily replaced.

As defined in section 1.5, the first step of this thesis work is the definition of a formal model for an interactive multimodal information retrieval system accepting multimodal queries and accessing heterogeneous multimedia information sources. The main properties of an IMIR system are:

- ***Multimodality:*** the system deals with documents or information in different modes (text, audio, image, video, etc.) both in the query and in the retrieved documents. For instance, an IMIR system about health can integrate knowledge about diseases, clinical notes, patient opinions, etc. from different sources.

- ***Multidomain:*** the domain is defined as the topic which the system works in. It is multidomain when it encompasses information from more than one domain simultaneously. For example, a system that retrieves information from injuries in sports can be considered multidomain, even more if it complements the injuries information with treatments, other sport player suffering the same injuries, typical sports that can cause the same injuries, etc.

- ***Multilinguality:*** the system handles multiple languages, both on the query and collection of documents to be indexed.

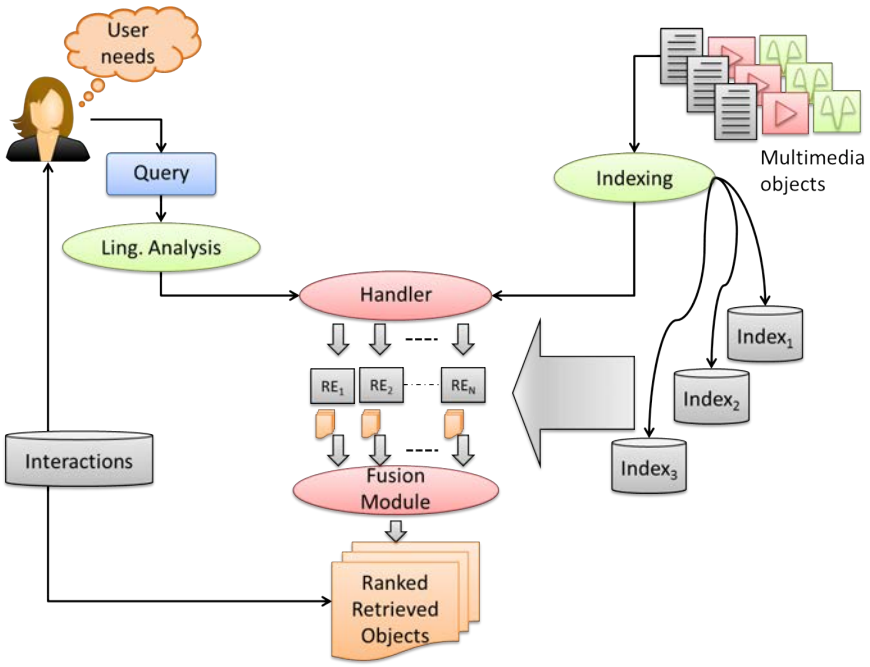


Figure 3.1: Processing flow of an IMIR system.

The processing flow of an IMIR system is an adaptation of the processing flow of a simple (non-multimodal) IR system. This flow is shown in figure 3.1. It begins

when users send the query to the system. It performs an analysis over the query and determines some relevant information about it: type (for example, if it is a question, or an image or a set of keywords), linguistic or semantic information (for example, if it is a name or verb, if it is some entity such as a person's name, city, etc.), low-level features (for example, the histogram of color or texture in an image), etc. Then, the query and these features are sent to the *'handler'*. This module uses the added information to determine the different REs that must be requested and in which order (if there is any order). After requesting all the REs, the handler returns a set of results' sets, i.e. a set containing all the lists of results of each RE that has been requested (or empty/null if it returns no results). Finally, this set of sets is analyzed in the *Fusion module* that combines, filters and reorders the results to get only a single final set of results. The interactivity is included by registering every action that users perform within the system (click in a document, relevance judgments, etc.).

In order to include this whole functionality in the formal model, the model has to fulfill a set of requisites that condition the definition of several components:

R.1. The different information and search modalities that will be studied and included.

R.2. The different retrieval techniques (engines) that can be considered within the model.

R.3. The way in which the engines are requested: *engine selection strategy*.

R.4. The fusion and reordering of the heterogeneous results.

R.5. The interactivity of the system.

3.2 Model Components

The six main parts comprising an IMIR system (Multimedia Information, Retrieval Engines, Query Modalitites, Handler, Results' Fusion and Interactivity) are explained in the following sections. Figure 3.2 shows a general architecture encompassing the elements that are considered in the formal model. The user interacts with the system through the interface (HTML/HTTP) where the interactions are recorded. The query is received by the handler that decides which retrieval engines are requested. These

engines return objects (information) which are sent to the fusion module. It is in charge of combining the results in a single set which is returned to the user.

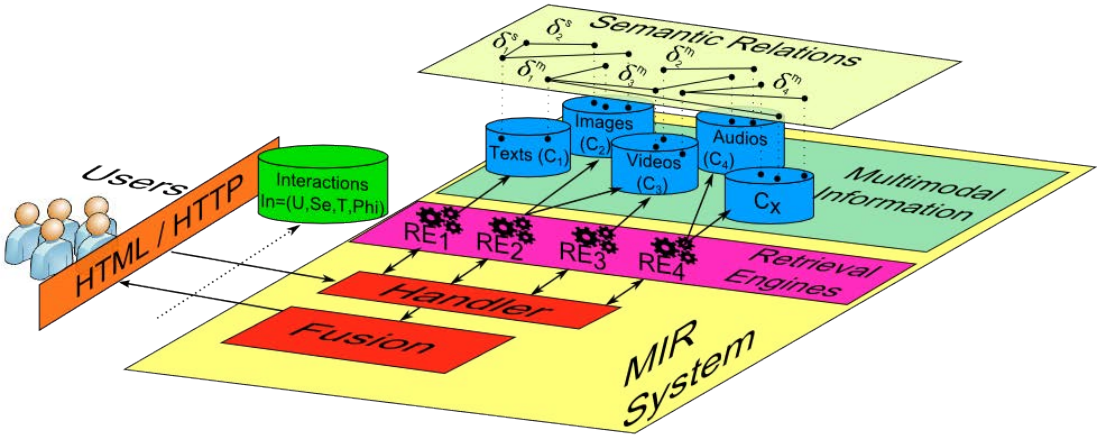


Figure 3.2: General architecture encompassing the components that are considered in the formal model to define MIR systems (Multimedia Information, Retrieval Engines, Query Modalitites, Handler, Results' Fusion and Interactivity)

3.2.1 Multimodal Information

Information (mono or multimodal) is the main component of an IR system. If the retrieved information (all documents) contains only one mode, such as text, image or video, information retrieval is monomodal. On the contrary, if it returns information in more than one mode, it is multimodal IR.

Multimodal information is sorted into a set of collections (see equation 3.1) where N is the number of collections.

$$\mathcal{C} = \{C_1, C_2 \dots C_N\} \tag{3.1}$$

Each collection (see equation 3.2) is composed of a set of documents where i represents the i^{th} collection and M the number of documents of the i^{th} collection.

$$C_i = \{D_{i1}, D_{i2} \dots D_{iM}\} \tag{3.2}$$

Each document (see equation 3.3) consists of a set of elements where P represents the number of elements of document D_{ij} and each element d_{ijk} is a multimedia element

(text, audio, image or video). That is, a document could be composed of elements of different modes.

$$D_{ij} = \{d_{ij1}, d_{ij2}, \ldots d_{ijP}\} \tag{3.3}$$

The model considers multimodal retrieval, so the information accessed has to be multimodal. The modality of a collection ($\mathcal{M}(\mathcal{C})$) is defined by the documents that compose it. The mode of documents ($\mathcal{M}(D)$) is defined by its elements, being monomodal when all its elements have the same mode or multimodal when there are at least two elements that have different mode (see equation 3.4).

$$\mathcal{M}(D) = \begin{cases} mono\ \forall i,j\ \mathcal{M}(d_i) = \mathcal{M}(d_j) \\ multi\ \exists i,j\ \mathcal{M}(d_i) \neq \mathcal{M}(d_j) \end{cases} \tag{3.4}$$

where $1 \leq i,j \leq K$ and $\mathcal{M}(d_i) \in txt, img, vid, aud, conc, trip, inst$ (completely described in section 4.2.6).

As an example we present a set of five documents from Wikipedia collection (C_W) (see figure 3.3).

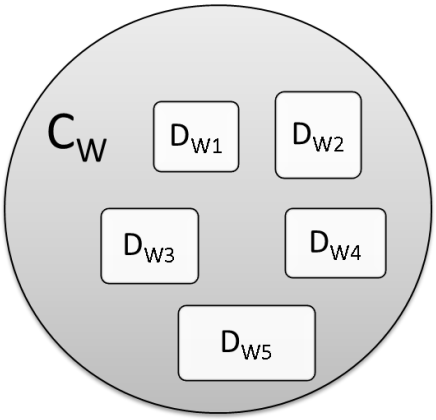


Figure 3.3: Wikipedia example collection containing five multimodal documents.

Let's assume that each document is composed of three elements: a text and two images. Figure 3.4 shows the first document of the collection (D_{W1}).

The formal definition of this example collection is shown in Table 3.1.

Besides their content, multimedia elements can be annotated with semantic information. Assuming that two documents containing related information have a relationship

Figure 3.4: Example of Wikipedia document where its elements are a text (d_{11}) and two images (d_{12} and d_{13}).

$$C_W = \{D_{W1}, D_{W2}, D_{W3}, D_{W4}, D_{W5}\}$$
$$D_{Wj} = \{d_{Wj1}, d_{Wj2}, d_{Wj3}\}\ \forall j \in [1,5]$$
$$\text{where } \mathcal{M}(d_{Wj1}) = txt \text{ and } \mathcal{M}(d_{Wj2}), \mathcal{M}(d_{Wj3}) = img.$$

Table 3.1: Formal description of Wikipedia example collection ($W = wikipedia$).

between them, both documents can be interconnected by semantic relations. Considering two documents (D_{ij} and D_{xy}), there are two types of semantic relations that can appear between them:

1. A **multimedia relation (δ^m)** relates directly two different documents (or multimedia elements) and is represented as $\delta^m(D_{ij}, D_{xy})$. This relation means that the two documents are related by δ^m. Two examples of multimedia relations present in the Wikipedia example are:

   - $\delta^m_1 = isPartOf \Rightarrow isPartOf(d_{W13}, d_{W32})$. This relations means that element 3 of Wikipedia document 1 is part of element 2 of Wikipedia document 3. Imagine that d_{W32} is an image of the most important scientist of the history

(including *Alan Turing*) and d_{W13} is a piece of the same image showing only *Alan Turing* (see figure 3.5).

Figure 3.5: Two images related by a multimedia relation in the ontology. d_{W32} is an image of the most important scientist of the history and d_{W13} shows only *Alan Turing*

- $\delta_2^m = summarize \Rightarrow summarize(d_{W51}, d_{W21})$. This relations means that element 1 of Wikipedia document 5 summarizes element 1 of Wikipedia document 2. In this case, d_{W51} could be the synopsis of a book and d_{W21} could be the complete content of the book (or a bigger part).

2. A **concept-based relation** or **semantic relation (δ^s)** relates two documents indirectly through a semantic concept. An indirect relations is defined by a connection between two documents in a graph through other nodes (semantic concepts in this case). A document is related to a concept represented as $\delta^s(D_{ij}, o)$ where o is a concept of the knowledge-based system. If two documents are related to the same concept ($\delta^s(D_{ij}, o)$ and $\delta^s(D_{xy}, o)$), then there is an indirect relation between both documents.

   - $\delta_3^s = mentions \Rightarrow mentions(d_{W11}, 'AlbertEinstein')$ means that element 1 of document 1 in Wikipedia collection mentions the concept 'Albert Einstein'.
   - $\delta_4^s = shows \Rightarrow shows(d_{W12}, 'AlanTuring')$ means that element 2 of Wikipedia document 1 shows the concept 'Alan Turing'.

These semantic relations are represented and stored in a knowledge-based system, such as an ontology or a concepts graph [Rotella et al., 2013].

Figure 3.6 shows a part of the semantic information associated with the Wikipedia example collection. As can be seen, four concepts and three documents are displayed. Document D_{W1} relates to the *Theories* and *Alan Turing* concepts, while document D_{W2} is related to two concepts: *Albert Einstein* and *Alan Turing*. The relationships between Alan Turing and documents establishes an indirect relationship between documents D_{W1} and D_{W2}. In addition, a direct relationship between documents (D_{W2} and D_{W3}) is observed where D_{W3} is part of D_{W2}.

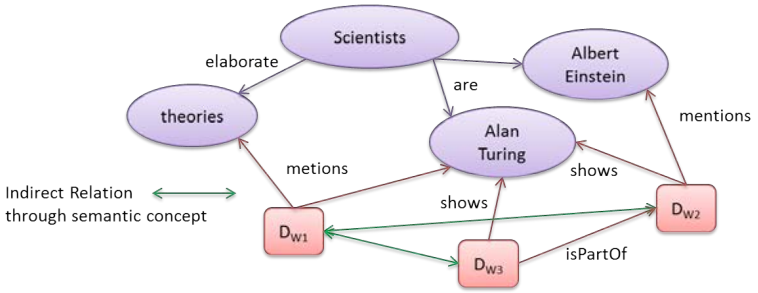


Figure 3.6: Partial view of semantic relations associated to the Wikipedia example collection.

3.2.2 Query Modalities

The query is the expression of the information need of the user materialized so that an automatic system is able to understand it. The way that users can express this information need has shifted from traditional keyword queries. Currently, systems allow more complex queries including multimedia elements or queries expressed using natural language.

The model considers multimodal queries that are defined as a set of elements

$$Q = \{q_1, q_2, \ldots, q_K\} \tag{3.5}$$

where:

- K is the number of elements of query Q
- each element q_k is a multimedia object element (text, audio, image or video)

The modality of the query ($\mathcal{M}(Q)$) is defined by its elements as in equation 3.4. Table 3.2 shows three examples of multimodal queries.

Query	Formal Representation	Textual Description
Crimea	$Q_1 = \{q_{11}\}$ where $\mathcal{M}(q_{11}) = txt$	You want to find information about Crimea, either news about the crisis between Russia and Ukraine, tourist information, etc.
Who is performing in this video? 	$Q_2 = \{q_{21}, q_{22}\}$ where $\mathcal{M}(q_{21}) = txt$ and $\mathcal{M}(q_{22}) = vid$	You want to find information of the artist performing in this video such as name, biography, concerts, etc.
 [76]	$Q_3 = \{q_{31}\}$ where $\mathcal{M}(q_{31}) = img$	A user has to go to a party and has seen a hairstyle that she likes in a picture (query). Using this image she wants to find similar hairstyles to get ideas for her hairstyle.

Table 3.2: Multimodal query examples, formal representation and description

3.2.3 Retrieval Engines

Information retrieval is the process of matching the query against the representation of the documents. It returns an information set ($\mathcal{S}$) (usually documents or parts of them) in any way related to a query (Q) posed by a user. The relationship between

[76] Image taken from `http://b2binformation.blogspot.com.es/2012/09/women-hair-styles-capable-of-adding-to.html` at 23/07/2015

the query and the documents (of the collections) will determine the retrieval technique to be used: query keywords in the document, similarity of low-level features (color or texture, frequency or movements) or equivalent semantic elements in query and documents. There are a wide variety of techniques that can be applied (see section 2.3).

The result of this retrieval ($\mathcal{S}$) is a set of documents (information) which are usually organized in a sorted list, but there are other approaches that return sets of documents (with no particular order) or sets of elements such as terms or semantic concepts.

The idea is to represent all these possibilities (as generally as possible). Because of that a retrieval engine (RE) is considered as a process ($\mathcal{P}$) that accesses some collections with a query and obtains a results' set from them. Equation 3.6 shows the triplet that represents a RE.

$$RE = [\mathcal{C}, Q, \mathcal{P}] \tag{3.6}$$

where:

- $\mathcal{C}$ represents a set of collections (explained in section 3.2.1) that is accessed to retrieve information.
- Q represents the query received at the input (explained in section 3.2.2).
- $\mathcal{P}$ is the retrieval approach. The retrieval approach is included in the model in order to generalize it although RE could be used as *'black-boxes'*. When a RE is considered a *'black-box'* it means that only the input and output of the RE is important, i.e., the way it works (its functionality) is not a relevant part of the study. Using RE as 'black-boxes' allows a system to encourage the flexibility of using RE defined by third parties and the re-usability of their own created RE in other works.

The RE functionality is defined as

$$\mathcal{S} = \mathcal{P}(Q) \tag{3.7}$$

where $\mathcal{P}(\bullet)$ represents the retrieval approach of the engine and $\mathcal{S}$ represents the result set when Q is sent to the engine. This retrieval engine output is a set of results

$$\mathcal{S} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots \mathcal{R}_L\} \tag{3.8}$$

where L is the number of results and each result $\mathcal{R}_l$ is composed of a set of multimedia objects (r_{li})

$$\mathcal{R}_l = \{r_{l1}, r_{l2}, \ldots r_{lT}\} \tag{3.9}$$

where:

- L is the number of results being $1 \leq l \leq L$.
- r_{li} is each multimedia object of result l
- T is the number of components of result $\mathcal{R}_l$.

Some examples of retrieval techniques are enumerated next.

1. Term frequency - Inverse document frequency (TF-IDF) [Aizawa, 2003] retrieves a list of textual documents ranked by the inverse frequency of the terms of the query that are present in the resulting document. The documents are indexed characterized by the terms present in them and the matching is done by comparing the terms of the query and the terms of each document. The documents are represented using known models such as probabilistic model [Jones et al., 2000], vector space model [Salton et al., 1975] or boolean model [Cavanagh, 1976] in order to compare them.

2. Content-based image retrieval (CBIR) [Smeulders et al., 2000] retrieves a set of images by their low-level features. The characterization of the documents is done by analyzing them in order to extract their low-level features (for example, texture, color, shape, etc.). The query is also analyzed and the matching between query and documents is performed comparing the low level features between the query and the documents (images) of the collection.

3. Semantic Search retrieves a set of documents by matching the semantic concepts of the query and the documents [Medina-Ramírez, 2007; Shah et al., 2002; Worring et al., 2007]. The documents are processed to extract the semantic concepts they contain (if these concepts are not explicitly given) and then the concepts are indexed. The concepts of the query are extracted and they match the index to retrieve relevant documents.

3.2.4 Managing multiple retrieval engines by a handler.

Multimodal information retrieval is not limited to request a single multimodal source, but requesting various monomodal sources (with different modes) is also considered as multimodal retrieval. Querying several sources is more appropiate due to the current distribution of Web content[77]. This is based on the fact that many websites are specialized in certain types of content (Youtube[78] for videos, Flickr[79] or Instagram[80] for images, Spotify[81] or SoundCloud[82] for audio, Google[83] or Yahoo[84] for text although they are also working with other modes). The problem arises when deciding which of every available source is requested with each query.

The model names this module as *handler* ($\mathcal{H}$) and defines it as a triplet (see equation 3.10).

$$\mathcal{H} = [\mathcal{E}, Q, \Xi] \tag{3.10}$$

where:

- $\mathcal{E}$ represents a set of REs normally defined as an ordered list.
- Q represents the input query.
- Ξ is the handling strategy. The handling strategy is in charge of selecting which REs are requested and in which order for the current query.

[77] http://www.triblio.com/blog/justify-doubling-content-marketing-budget-6-steps/ at 23/07/2015
[78] https://www.youtube.com/
[79] https://www.flickr.com/
[80] http://instagram.com/
[81] https://www.spotify.com/es/
[82] https://soundcloud.com/
[83] https://www.google.es/
[84] https://es.yahoo.com/

The functionality of the handling strategy (Ξ) is to provide a ranking of the available *REs* depending on the query (see equation 3.11).

$$\mathcal{E}' = \Xi(\mathcal{E}, Q) \tag{3.11}$$

where $\mathcal{E}$ is the complete set of available retrieval engines and $\mathcal{E}'$ is the subset selected for being retrieved by query Q.

This model is particularized for defining the handling strategy by a set of rules (the structure of a rule can be seen in equation 3.12).

$$conditions \rightarrow \mathcal{E}' = \{RE_1, \ldots, RE_Z\} \tag{3.12}$$

where $\mathcal{E}' = \{RE_1, \ldots, RE_Z\}$ represents the set of retrieval engines (typically an ordered list) that are requested if *'condition'* is met and N is the number of available *REs* ($Z \leq N$).

The handling strategy (Ξ) can be executed in parallel, sequential or hybrid ways as it is explained below.

Parallel Execution

The handler decides which *REs* are triggered and sends the query (or part of the query) to them at the same time. The results' set coming from each *RE* is then sent to the fusion module component (see section 3.2.5). A schema of parallel execution is shown in Figure 3.7.

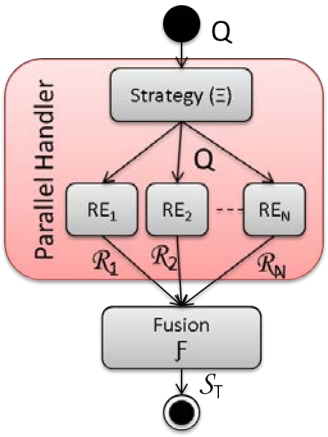


Figure 3.7: Parallel execution of several *REs*

Works based on web search engines such as Sushmita [2012] (using Yahoo!) or Malla et al. [2011] (using Bing), where each 'vertical' is requested at the same time and with the same query (expressed by the user), are clear examples of parallel execution.

Sequential Execution

In sequential execution the different *REs* are requested in an ordered way. As defined in Galiano [2011], there are two types of sequential execution: (1) filtering, where a RE only retrieves results from the results previously defined as relevant by other REs; and (2) feedback, where information extracted from the most relevant results of a RE is used for modifying the query sent to the next RE. A schema of sequential execution can be seen in figure 3.8.

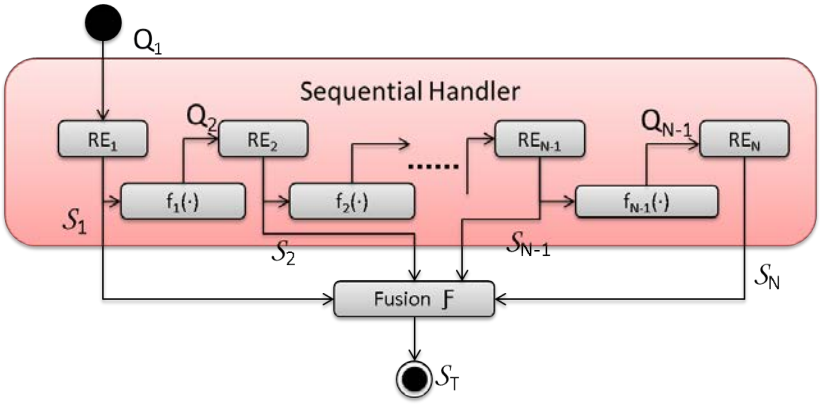


Figure 3.8: Sequential execution of several *REs*

Each query is a function of the previous queries and the results' set of the previous *RE* (see 3.13).

$$Q_x = f_{x-1}(Q_{x-1}, \mathcal{S}_{x-1}) \tag{3.13}$$

where:

- $x \geq 2$
- $\mathcal{F}(\bullet)$ is the results' combination function.
- $\mathcal{S}_T$ is the final results' set obtained from the fusion of the retrieval engines' output $\mathcal{S}_T = \mathcal{F}(\mathcal{S}_1, \mathcal{S}_2 \ldots \mathcal{S}_N)$.

- $\mathcal{S}_{x-1}$ is the results set from RE_{x-1}.
- Q_{x-1} is the query used for requesting RE_{x-1}.
- Q_x is the new query which is used for requesting RE_x.
- $f_{x-1}(\bullet)$ represents the function used to combine query Q_{x-1} and results $\mathcal{S}_{x-1}$ in order to create a new query Q_x.

This type of execution can be found in semantic search systems, where an extraction of semantic concepts is performed over the query, for later using this concepts as query to request a text-based or a concept-based engine [Worring et al., 2007]. In this case, the query is completely changed for using concepts. An example is shown in figure 3.9. The input query (Q_1) contains a set of tokens from which two are concepts, that are used later for generating the query (Q_2) of the concept-based IR.

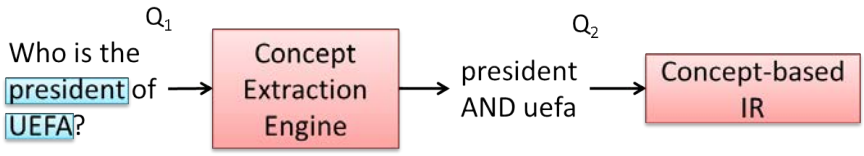


Figure 3.9: Example of sequential execution of two RE: a concept extraction engine and a concept-based IR

Hybrid Execution

Hybrid execution is a mix of parallel and sequential execution. There is a main pipeline execution as in sequential execution, but instead of executing one RE at each step, there are a set of REs that are executed in parallel. A schema of hybrid execution of requesting different REs is shown in figure 3.10.

Each query is a function of the previous query and the previous final results' set (see equation 3.14).

$$Q_x = f(Q_{x-1}, \mathcal{S}_T) \tag{3.14}$$

where:

- $\mathcal{F}(\bullet)$ is the results' combination function.

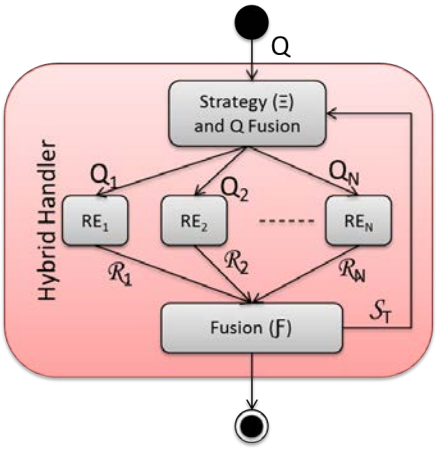


Figure 3.10: Hybrid execution of several $RESs$. Q_i $\forall i \in [1, N]$ are the elements of Q that are sent to each retrieval engine.

- $\mathcal{S}_T$ is the final results' set obtained from the fusion of the retrieval engines' output $\mathcal{S}_T = \mathcal{F}(\mathcal{S}_1, \mathcal{S}_2 \ldots \mathcal{S}_N)$.
- Q_{x-1} is the query used for requesting RE_{x-1}.
- Q_x is the query used for requesting RE_x.
- $f(\bullet)$ represents the function used to combine query and results in order to create a new query.

An example of hybrid execution is the voice search used by Siri[85] or Android voice transcription[86]. First of all, they transcribe the received audio and later they use the obtained text as query to request the corresponding information retrieval systems (full text search, question answering systems, etc).

3.2.5 Managing results from several retrieval engines: results' Fusion

Requesting more than one RE leads to having heterogeneous results (different mode, content structure, etc). The fusion module receives a set of results' sets ($\mathcal{S}_x$ $1 \leq x \leq N$), each one coming from a different retrieval engine. Because of that, this module has to

[85] http://www.apple.com/es/ios/siri/ accessed at 23/07/2015

[86] http://www.google.com/insidesearch/features/voicesearch/index-chrome.html accessed at 23/07/2015

deal with these results in order to combine, filter and re-rank them into a single results' set ($\mathcal{S}_{final}$).

A result is represented by a pair document-score ($\mathcal{R}_{xy} =< D_{xy}, \gamma_{xy} >$). The results obtained from each RE must be combined in order to get a single results' set. To perform this aggregation, a lineal combination [Strang, 2006] is used. This lineal combination compute the final score of a document as the weighted sum of the scores that each retrieval engine returns for this document.

The results' set of a certain RE is defined as a vector, which contains a set of pairs document-score for each document existing in the target collections. If the retrieval engine has not returned a certain document as relevant, i.e., the document is not present in its results' set, then this document gets a score of zero. Considering these requirements, the final results' set could be formally defined as a matrix product (see equation 3.15).

$$\mathcal{S}_{final} = \mathcal{A} \cdot \mathcal{V} \tag{3.15}$$

where:

- $\mathcal{A}$ represents a vector containing the weight coefficients of each RE.

$$\mathcal{A} = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_N \end{bmatrix} \tag{3.16}$$

  where α_n represents the weight of the results' set of the n^{th} RE.

- $\mathcal{V}$ represents a matrix containing the results' scores. Each column corresponds to a concrete document of the collections and each row corresponds to a retrieval engine. The intersection of a row and a column stores the score assigned to the document by the retrieval engine (RE).

$$\mathcal{V} = \begin{bmatrix} \mathcal{S}_1 \\ \vdots \\ \mathcal{S}_N \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \cdots & \gamma_{1M} \\ \vdots & \ddots & \vdots \\ \gamma_{N1} & \cdots & \gamma_{NM} \end{bmatrix} \tag{3.17}$$

  where:

  - N is the number of REs.

  - M is the size of the possible results vector (see equation 3.20): this size is the sum of the sizes of all the collections avoiding repeated documents.
  - γ_{ij} represents the score of result $\mathcal{R}_{ij}$ (the result number j of the i^{th} RE).

- $\mathcal{S}_{final}$ represents the final results' scores (generally an ordered list).

$$\mathcal{S}_{final} = \begin{bmatrix} \gamma_1^{final} \\ \gamma_2^{final} \\ \vdots \\ \gamma_M^{final} \end{bmatrix} = \begin{bmatrix} \alpha_1 \cdot \gamma_{11} + \alpha_2 \cdot \gamma_{21} + \ldots + \alpha_N \cdot \gamma_{N1} \\ \vdots \\ \alpha_1 \cdot \gamma_{1M} + \alpha_2 \cdot \gamma_{2M} + \ldots + \alpha_N \cdot \gamma_{NM} \end{bmatrix} \tag{3.18}$$

  where γ_m^{final} represents the score of m^{th} result after combining every results' set of each RE.The score of a result is generalized as

$$\gamma_i^{final} = \alpha_1 \cdot \gamma_{1M} + \alpha_2 \cdot \gamma_{2M} + \ldots + \alpha_N \cdot \gamma_{NM} = \sum_{j=1}^{M} \alpha_j \cdot \gamma_{ji} \tag{3.19}$$

  where

  - N is the number of REs.
  - M is the size of the possible results vector (see equation 3.20): this size is the sum of the sizes of all the collections avoiding repeated documents.

This definition is only possible if the results' set of each RE ($\mathcal{S}_n$ for $\forall n$) has the same length and they contain zeros for each element of the collections that has not been retrieved by RE_n. This length is equals to the number of documents (avoiding repetitions) that all the collections contain. This implies that these vectors have a size defined in equation 3.20.

$$size(\mathcal{S}_n) = size(\bigcup_{j=1}^{N} \mathcal{C}_j) \tag{3.20}$$

where:

- N is the number of requested REs
- $\mathcal{C}_j$ represents the document collections used by RE_j.

To clarify the fusion process, an example is shown in figure 3.11. Suppose a IMIR system that uses two different retrieval engines (RE_1 and RE_2) where each engine uses a different collection of documents ($\mathcal{C}_1$ and $\mathcal{C}_2$ respectively). Each collection contains 5 documents and 3 of them are shared between both collections.

Each retrieval engine returns a results' set containing 5 results (scores). On the contrary, while applying the fusion strategy, each results' set is converted into an intermediary results' set (R1' and R2') that contains seven results (scores). The documents (from every collection) that have not been retrieved by a RE are assigned a score of 0. Then, these two intermediary results' sets (R1' and R2') are fused using equation 3.18. The final results' set will have a size of 7 results, although each RE will return only 5 scores (for the documents they are returning).

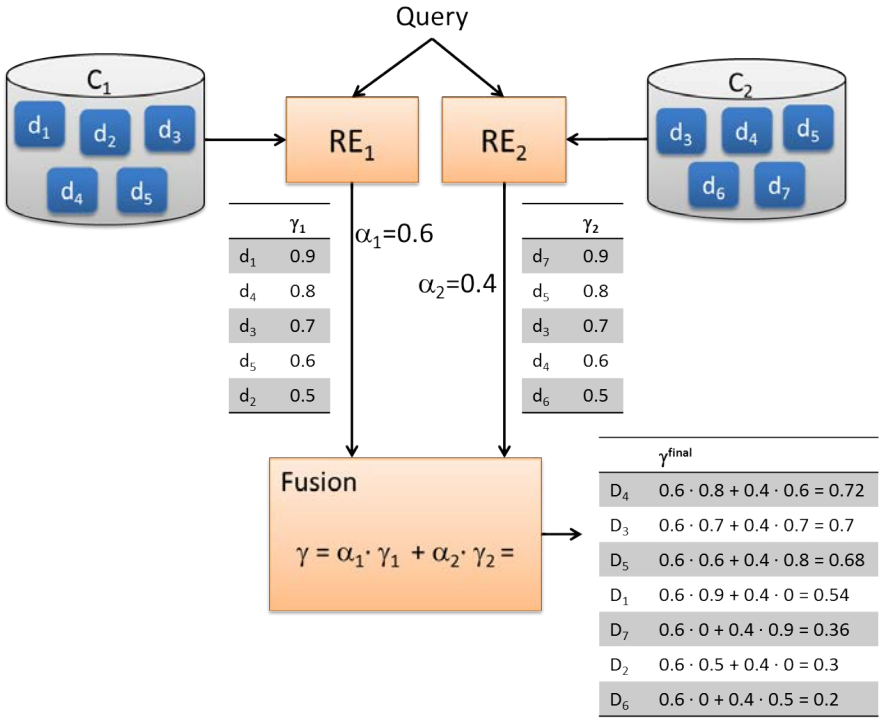


Figure 3.11: Example of fusion algorithm when two REs are requested, each one returning five results and obtaining finally a results' set of final size equal to seven.

3.2.6 User Interactions

In an interactive system the user is part of the system and users activity (queries, displayed results, timestamps, etc.) are recorded. The information logged from the users is very different depending on each application or system. The model definition of an interaction must consider the registration of as much different information from the user as possible.

An example of this kind of recording are the 'Cookies' in web browsers. A 'cookie' is a *'piece of information sent by a web site and stored in the user's browser so that the website can consult the previous user activity'*[87].

The first assumption of the model is that *users* are registered and they perform interactions during *sessions*. A session is defined by an initial and a final timestamp and consists of a set of interactions, since the user enters the system until it disconnects. Similarly, the interactions are organized in terms of the user who made them. With these considerations equation 3.21 shows a quintuple representing an interaction (In).

$$In = (U, Se, ts, \mathcal{T}, \Phi) \tag{3.21}$$

where:

- U is the identifier of the user who did the interaction.
- Se refers to the session in which the interactions was registered that is characterized by the initial and final timestamps.
- ts is the timestamp when the interaction (user's action) has been done.
- $\mathcal{T}$ is the type of the interaction. Again, the type of interactions that can be registered depends on the system and must be specified at implementation time. Some examples could be: *login*, *logout*, *search*, *clikOn*, *View*, etc.
- Φ is a field that has been defined in order to allow adaptability. The information stored in this attribute can be different for each interaction type.

  For example, in case that a user logs into the system, this parameter does not store anything and has a null value ($NULL$). On the contrary, if a user performs

[87] Taken from http://es.wikipedia.org/wiki/Cookie_(inform%C3%A1tica). Accessed at 23/07/2015.

a search, the text of the query is stored (*'video goals Barcelona'*), while if the user displays a result, it stores the name of the result, the source it comes from, its position in the results' list and its mode (*'news008-qa-3-text'*).

Table 3.3 shows some examples of interactions and offers a textual explanation of the actions performed by the user.

Action's Description	Interaction
User *'jmschnei'* performs a **search** in the session with identifier *'session001'* at the moment *'timestamp'* using the textual query *'video goals Barcelona'*.	`(markus, session001, '18-07-2013 11:01:24', search, 'video goals Barcelona')`
'User034' has **visualized** a result with *id='news008'* from the source *'qa'* that was at position *'3'* of type *'text'* at the moment *'29-11-2013 09:51:04'*.	`(user034, session288, '29-11-2013 09:51:04', visualizacion, 'news008-qa-3-text')`
'David' has marked as relevant a result with *id='img147'* from the source *'ont'* that was at position *'5'* of type *'img'* at the moment *'12-08-2013 22:09:26'*.	`(david, session004, '12-08-2013 22:09:26', exploration, 'GOOD-img147-ont-5-img')`
'Viktor' has logged in the system creating the session *'session562'* at the moment *'04-04-2013 13:14:45'*.	`(viktor, session562, '04-04-2013 13:14:45', login, NULL)`

Table 3.3: Examples of user actions and the interactions that are generated according to equation 3.21

3.3 Validation of the formal model.

The formal model has been validated by defining a fully functional prototype. Therefore, every feature of the formal model that is implemented in the prototype is considered as validated.

As said in Cabot and Clarisó [2014], *'the quality of the models can be regarded from many different perspectives. It is necessary to make sure that the models are realizable (i.e., the structural models should be able to be satisfied, the states in a behavioral model should be reachable, etc.)'.* With the implementation of the prototype we have demonstrate the capability of realization of a real system based on the model.

It is noteworthy that according to the definition of the model components, any items that have been defined in the prototype may be exchanged for other elements also defined following the model. This is an important advantage of defining components based on a model.

The prototype retrieves multimodal information (text, video and image) from sports and news domains (see chapter 4 for a detailed description). The elements of the model that have been implemented in the prototype that is fully described in chapter 4 are:

- The prototype accepts three types of **multimodal queries**: textual query, voice query and a combined query composed by a text and an image. The format of the textual query can range from keywords or natural language text to ontology concept names. For example, the query `http://www.buscamedia.es/ontologies/M3/logo/FC_Barcelona` can be used for obtaining the information related to the ontology concept *'FC_Barcelona' = Football Club Barcelona.* This type of queries are, normally, not generated by users, but when a natural language query returns ontology concepts as results, these concepts can be used for performing exploratory search using it as query.

- Several modes of information are retrieved by the prototype: text (news, metadata from images and videos), images and videos. Although three possible modes represent a limited range, it proves that **multimodal information** can be defined and handled.

- It is required that there are several retrieval engines to fully verify that the handler is working. Because of that, three retrieval engines are included in the prototype (see section 4.2.4).

- A **retrieval engines handler** is implemented by a rule-based approach. That is, rules specify according to query characteristics which retrieval engines are requested and in which order (if necessary). The decision taken by the rules depends on the characteristics of the query. In this case, we are using the linguistic characteristics such as type (question, more than three tokens, keywords, etc.), number of entities, number of verbs, size (in number of tokens), etc.

- The results from the retrieval engines are fused by a round-robin strategy. This strategy has been formally defined following the model (section 4.2.7). So, it demonstrates that the definition of a **results fusion module** following the equations of the formal model is possible.

- All actions performed by users are recorded by the prototype. These **interactions** are logged according to the format set by the model and stored in the database explained in section 6.3.

- This prototype uses **semantic knowledge** for searching the ontology (see section 4.2.4). Semantic information is also used for results visualization in the semantic clusters view. This visualization shows the ontology concept results (if they have been retrieved) grouped by their semantic categories (see section 4.2.8).

4

Development of an IMIR prototype in sports domain

Having defined a formal model, the next step is to develop a prototype where the concepts of the model are applied to test whether these concepts are applicable to solve a real problem. The objective of the prototype developed is twofold:

1. The validation of the model by demonstrating that this system is fully functional.
2. The creation of a complete system in the framework of a research project called Buscamedia.

Furthermore, the purpose of this thesis is to adapt the behavior of a multimodal IR system, so we need a set of user interactions that are related to the collections of documents that we use. Those collections of documents have been imposed by the Buscamedia project in which we collaborated and they are described in detail in the section 4.2.1. The problem is that the 'standardized' sets of interactions (see section 2.7) work mostly with documents' collections that are not semantically related. Therefore, the easiest way to agglomerate a set of interactions is to create an IR prototype that retrieves information from the Buscamedia project collections, launching a user evaluation and recording the interactions.

The model defined in section 3 details a complete multimodal IR system. The prototype has been defined using a subset of model components: multimodal information,

multimodal query, multiple retrieval engines, handler, results' fusion and interactions management. The full description of these elements and their concrete definition using the model is made in section 4.2.

4.1 Research context

Part of this thesis was carried out during the collaboration in Buscamedia project. **BUSCAMEDIA**[88] was a CENIT project that aims to achieve significant progress in the areas of semantics, audiovisual production and distribution of rich media regardless of consumer networks and terminals, with the aim of creating a single semantic multimedia search engine.

The aim of Buscamedia project is the development of multimedia search technologies and automated resource management to develop an audiovisual ecosystem. This ecosystem will allow, in the future, the creation of new products, processes or services, and integration of technologies of strategic interest besides exploitation of their contributions to Spanish-speaking markets.

The main contributions of this project were: *'putting the Spanish industry to the head of state of the art search systems and multimedia production and audiovisual automation processes, supporting innovation in these technologies for the development of ontologies based on the Spanish semantics, and to serve the basis for audiovisual own classification'*. Linguistic developments have been developed in all official languages of the Spanish state.

The partners taking part in the project can be classified into three types:

- **Distributed systems experts**
  - *Atos SE (Societas Europaea)* is an international service company information technologies.
  - *GFI Informática* is a Consulting and IT Services company.
  - *Indra* as the second Spanish company in R&D&i.
  - *Fractalia* is a Spanish R&D&i company leader in the development of robust and effective remote management and control of large networks of systems.

[88] http://www.cenitbuscamedia.es/ accessed at 23/07/2015

- **Software providers**
  - *DAEDALUS* is a company with extensive experience in research, innovation and technology transfer in the field of Language Technology.
  - *Bilbomática* is a consulting and computer services expert in annotating and semantic indexing of multimedia content.
  - *Barcelona Music and Audio Technologies (BMAT)* is a technology-based company specializing in the field of products and services related to digital music.
  - *Ingeniería y Sistemas de Información y Documentación (ISID)* is a Spanish company specializing in developing multimedia software management solutions, also called Rich Media: video, audio, images.
  - *iSOCO S.A.* is a leader in the development and commercialization of semantic web technologies inside and outside the borders of our country.
- **Content providers**
  - *La Corporació Catalana de Mitjans Audiovisuals (CCMA)* is a public agency that manages broadcasting and television services of the Generalitat de Catalunya.
  - *Televisió de Catalunya (TVC)* is an organization of production and television broadcast.

Each company was accompanied in the project by a research center. In our case, we (University Carlos III of Madrid - UC3M) were working with **DAEDALUS**[89].

The development of this thesis uses the collections of documents that have been generated in this project, called 'Sports20'. This collection is described in detail in section 4.2.1.

4.2 Prototype Description

The prototype has been implemented fitting the model described in chapter 3. The architecture is shown in figure 4.1 where seven parts are clearly differentiated: multimodal

[89] `http://www.daedalus.es/` accessed at 23/07/2015

collections, semantic resources, multimodal query, retrieval engines (REs), handler, fusion module and graphical user interface (GUI).

As figure 4.1 shows, the processing flow of the prototype follows the basic functionality of an IR system. The process begins when a user sends a query to the system (through the GUI). The query is then sent to the rule-based handler which analyzes the query and determines its type (text, audio or a combination of text and image). The handler is in charge of requesting the available retrieval engines (depending on the query and its type). Each retrieval engine returns a set of results to the handler which sends them to the results' fusion module. This module combines, filters and reranks the results to obtain a single results' set. Finally, this single results' set is returned to the user.

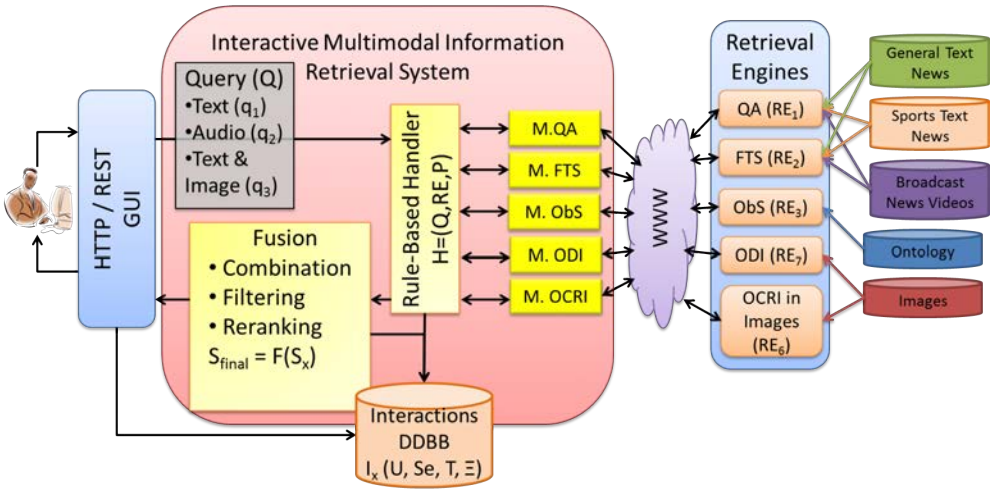


Figure 4.1: Architecture of the prototype

The main parts comprising the IMIR system are deeply described in the following sections.

4.2.1 Multimedia Collections

Information retrieval systems need collections of documents to retrieve information, but they also need collections of documents to evaluate the performance of the systems (see chapter 5). Each system is validated using a collection of documents, what was a problem to compare the performance of different systems evaluated with different

collections. The rise of evaluation forums had the goal to avoid these comparison problems by offering a collection of documents that every participant in the forum could use to evaluate its system. And since they all use the same collection of documents, it is easy to establish a comparison between the different systems, that is, the state of the art in a specific task.

There are several evaluation forums in which comparisons between systems through common document collections are made. These benchmarks are used to determine the goodness of a system within the current state of a research area. Some of the most popular forums in IR are TREC (Text REtrieval Conference) and CLEF (Cross-Language Evaluation Forum), where the following tracks are identified:

- TREC Interactive Track 2002: the high-level goal of the Interactive Track was the investigation of searching as an interactive task by examining the process as well as the outcome. It used an ad-hoc collection called .GOV (currently is not available) that followed the structure of web search results: title and content[90].

- TREC Web Track 2013: The goal of the TREC Web track is to explore and evaluate retrieval approaches over large-scale subsets of the Web. It used *'the 870-million page ClueWeb12 database, that consists of crawling the web for about 1 billion pages, web page filtering, and organization into a research-ready dataset'*[91].

- iCLEF (interactive track of Cross-Language Evaluation Forum): *'cross-Language search capabilities are studied from a user-inclusive perspective. A central research question is how best to assist users when searching information written in unknown languages, rather than how best an algorithm can find information written in languages different from the query language'*. It uses *'Flickr, a large-scale, web-based image database based on a large social network of WWW users, with the potential for offering both challenging and realistic multilingual search tasks for interactive experiments'*[92] *at 23/07/2015.*

These collections integrate multimedia objects (such as text content and image) but they are not useful for our purposes because they have no semantic relationships

[90] http://trec.nist.gov/data/t11_interactive/guidelines.html accessed at 23/07/2015
[91] `http://research.microsoft.com/en-us/projects/trec-web-2013/` accessed at 23/07/2015
[92] Taken from http://nlp.uned.es/iCLEF/

between documents. The definition of semantic relationships in a collection of documents means that there is a knowledge base that stores a series of semantic associations among documents (in the collection). Semantic relationships are defined in advance and define both document and concept levels (see section 4.2.2). The relationships between documents define direct relations between documents (a photo *belongs* or *is contained* in a video), while concept relationships define concepts found in multimedia elements (*showing* the concept or *mentioning* it, etc).

Since there is no available Spanish multimodal collection in which documents are semantically related and due to work performed in the Spanish Buscamedia project, the ad-hoc collection generated for this project was used. This collection is known as ***'Sports20'*** and is multidomain covering football, basketball and Formula One sports. It has been supplied by content providers partners. They were obtained during October 2010 and it is composed by four subsets of documents in different modes.

- The first sub-set is composed of 9245 textual news that have been compiled from various newspapers and most consist of title, subtitle and body: $C_1 = \{D_{1,i}\}$ where $1 \leq i \leq 9245$ and $\mathcal{M}(D_{1,i}) = txt$.

- The second sub-set encompasses 33 videos that are sports newscasts with an average duration of 3:51 minutes, the shortest being 1:23 and the longest 5:31 minutes: $C_2 = \{D_{2,j}\}$ where $1 \leq j \leq 33$ and $\mathcal{M}(D_{2,j}) = vid$. These videos contain a manually generated transcription.

- The third sub-set contains 659 images that were obtained by extracting key-frames of the videos: $C_3 = \{D_{3,k}\}$ where $1 \leq k \leq 659$ and $\mathcal{M}(D_{3,k}) = img$.

- The fourth sub-set is composed of 1191 semantic concepts obtained by semi-automatic population of an ontology (explained in section 4.2.2): $C_4 = \{D_{4,x}, D_{4,y}\}$ where $1 \leq x \leq 1191$ and $\mathcal{M}(D_{4,x}) = conc$ and $1 \leq y \leq 1590$ and $\mathcal{M}(D_{4,y}) = inst$.

4.2.2 Semantic Resources: Ontology

The prototype takes advantage of semantic search using a multidomain ontology. It is a knowledge-based system with a double functionality: it relates semantically the documents of the collections and it is a retrieval engine (see section 4.2.4).

This ontology [iSoco, 2013] is containing multilingual documents in three languages: two from Spain (Spanish and Catalonian) and English. It is composed of 30 smaller ontologies, having a total of 1191 classes, 722 object properties that relates them and 338 data properties. Besides, it has been populated with 1590 individuals. Figure 4.2 shows part of the football domain showing different classes, individuals and their relationships. This football ontology has information related to competitions (cups, leagues, divisions) and teams (stadium, players, etc.). In addition, it also includes information about players, time periods (duration of a match and duration of one season) and objects such as stadiums or items used in games (ball, goal, etc.). The idea is to use the ontology to model a domain as close to reality as possible, including the maximum number of details.

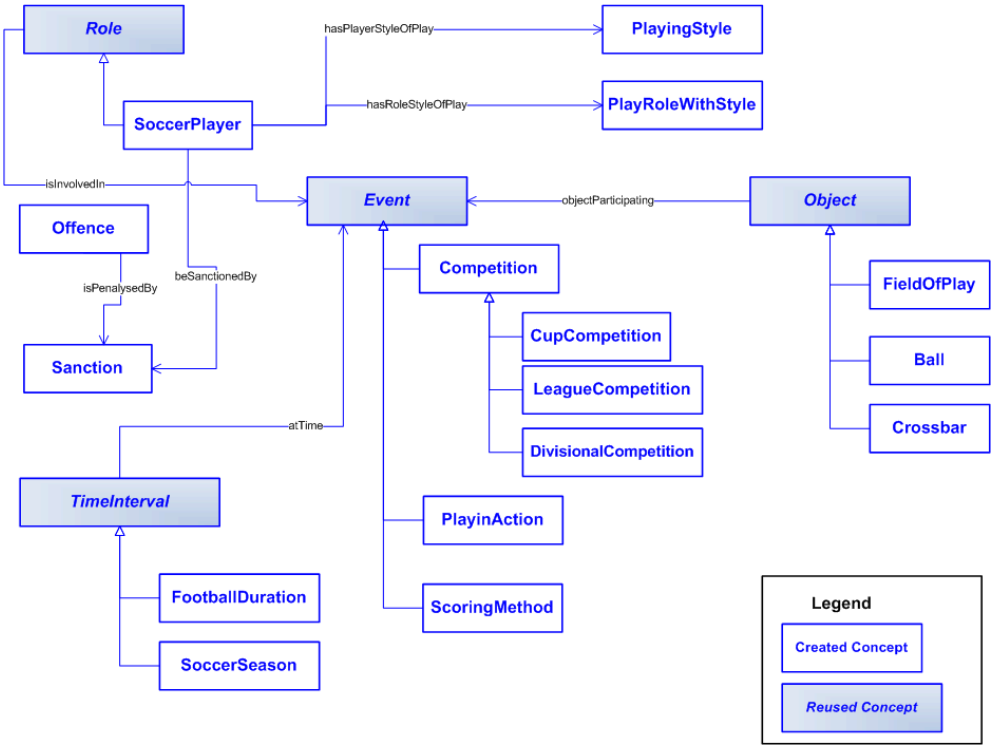


Figure 4.2: Sub-schema of football sub-domain of Sports20 ontology

The set of multimedia and semantic relations inside the different collections is important because it is used in browsing throughout results, i.e. the retrieved concepts (from the ontology) contain relations to other concepts or documents, and following

these relations new information can be obtained (by exploratory search).

The ontology contains 94 multimedia relations. Some examples are explained next and in table 4.1:

- δ_1^m = *'isVideoFragmentOf'* relates two video documents. For example, **isVideoFragmentOf**`('video4','video23')` determines that *'video4'* is part of *'video23'*.
- δ_2^m = *'isKeyframeOf'* relates an image with a video such as *'img32'* being a keyframe or fragment of *'vid23'* (**isKeyframeOf**`('img32','vid23')`).
- δ_3^m = *'isSourceOf'* or *'isRelatedTo'* connect two media resources (image with video or audio with video). For example, **isRelatedTo**`('image32','audio7')` declares a relation between *'image32'* and *'audio7'*.

Relation	Description
appearsIn	A multimedia element appears within the content of another multimedia element.
consistOf	Multimedia element 1 consists of part of multimedia element 2.
containImage	A multimedia element contains the referred image.
containVideo	A multimedia element contains the referred video.
isAudioFragment	A multimedia element (audio) is a fragment of another multimedia element (audio).
isMediaFragmentOf	A multimedia element is a fragment of another multimedia element.
isVideoFragmentOf	A multimedia element (video) is a fragment of another multimedia element (video).
presentsVideoShot	A multimedia element (image) shows a shot of a video.

Table 4.1: Representative examples of multimedia relations contained in the M3 ontology

The semantic relations are divided into two types: relations between a document and a concept and relations between two concepts. The ontology contains a total number of 1735 semantic relations. Some examples of relations between concepts are:

- **belongsToTeam**('Lionel Messi', 'FC Barcelona'): a concept referring to a football player belongs to a concept referring a football team.
- **playRole**('Lionel Messi', 'Football Player'): relation that defines the role of a concept. In this case it determines the role of concept 'Lionel Messi' being a 'football player'.

On the contrary, examples of relations between documents and concepts are:

- **isAbout**('video23', 'Football') : defines the topic of a multimedia element.
- **mentions**('video4', 'FC Barcelona') : defines that a multimedia element mentions a concrete semantic concept.
- **appearsIn**('Michael Phelps', 'image32') : determines that a certain concept is mentioned in a multimedia element.
- **refersTo**('audio7', 'Formula One') : determines that a multimedia elements refers to a certain topic or concept.

Relation	Description
appearsIn	A concept appears in a multimedia element.
exhibits	A multimedia element exhibits a concept.
hasDomain	A multimedia elements belongs to a domain specified in the ontology.
isAbout	A multimedia is about a topic determined in the ontology.
isRelatedTo	A multimedia element is related to a concept of the ontology.
mentions	A multimedia element mentions a specific concept.
shows	A multimedia element shows a specific concept.
refersTo	A multimedia element refers to a specific concept.
hasFormat / isFormatOf	A multimedia element has a specified format.

Table 4.2: Representative examples of semantic relations between an ontology concept and a multimedia object contained in the M3 ontology

Besides the semantic knowledge, the ontology works also as a retrieval engine with three different search functionalities (see section 4.2.4).

4.2.3 Query Modalities

The model allows the definition of any type of query (mode) but in this prototype there are three implemented query modalities[93]: textual query, voice query and the combination of text and image in a query.

- ***Textual query:*** the query is a text from only one token to a complete sentence.

$$Q_{text} = \{q_1, \ldots, q_i\} \tag{4.1}$$

  where $\forall i\ \mathcal{M}(q_i) = txt$. Some examples of this type of query are: *'Barcelona'*, *'last record from Fernando Alonso'* or *'In which team does Navarro play?'*.

- ***Voice query:*** the query is an audio file containing a spoken query.

$$Q_{voice} = \{q_1\} \tag{4.2}$$

  where $\mathcal{M}(q_1) = aud$. Once the spoken query has been transcribed, it is handled as a textual query.

- ***Textual and image query:*** the query is the combination of a textual query plus an image.

$$Q_{text-image} = \{q_1, \ldots, q_M, q_{M+1}\} \tag{4.3}$$

  where

  - $\mathcal{M}(q_i) = txt$ for $\forall i \in [1, M]$
  - $\mathcal{M}(q_{M+1}) = img$

  An example of this kind of query is *'When did the event in the image take place?'* together with the image in Figure 4.3.

[93] These three modalities were proposed under the project Buscamedia by the partners which were final users of the use cases.

Figure 4.3: Image included in the query example containing the text: 'Salamanca, this morning. Huge fear in El Helmántico when Miguel García collapsed'

4.2.4 Retrieval Engines

As collections and query modes, the definition of the retrieval engines (*REs*) is based on specifications of the Buscamedia project (see section 4.1). Moreover, every *RE* has been defined, designed and implemented independently by third parties, i.e Buscamedia project partners. The REs that were used are briefly explained because their design and implementation is out of the scope of this work. Our approach will use these REs as black boxes that receive an input query and generate a set of ordered results. The prototype makes use of five retrieval engines:

1. Question Answering Search (**QAS**) compares the query with the documents in the collections and extracts an answer from the most relevant. It returns a set of results containing concrete answers and documents supporting them. This engine retrieves information using SOLR-LUCENE [Smiley and Pugh, 2009]. In addition, it performs morphological tagging, syntactic analysis, named entity recognition, semantic tagging and classification of the query. The final answers are obtained by a process of answers extraction and re-ranking from retrieved documents. For the linguistic analysis its proprietary technology MeaningCloud[94] is used. MeaningCloud combines the most advanced technologies to provide a simple, powerful and affordable way to extract meaning from social media. It

[94]https://www.meaningcloud.com/es/ accessed at 23/07/2015

provides graphical interfaces to allow users to easily customize the system using their own dictionaries and models. It can be used in different languages: Spanish, English, French, Portuguese and Italian. This retrieval engine is formally defined in equation 4.4 (based on the formal model described in chapter 3).

$$RE_1 = (C_1, Q_{text}, \mathcal{P}_1) \tag{4.4}$$

where $\mathcal{P}_1$ refers to the retrieval approach of finding concrete answers from documents for a textual query.

An example of query for QA engine can be:

¿Cómo se llama el presidente de la UEFA? (*What is the name of the president of UEFA?*)	The user needs the name of the president, but documents containing the name will not be as relevant as the proper name.

2. Full Text Search (**FTS**)[95] works as keyword-based retrieval. It returns a set of textual documents that contain the query keywords. The engine uses BM25F [Pérez-Iglesias et al., 2009] with the same push factors for information retrieval, and implements an analysis of Snowball [Porter, 2001] available for each language in Lucene [McCandless et al., 2010] (removal of stop words, stemming and removal of special characters and punctuation). This retrieval engine is formally defined in equation 4.5 (based on the formal model described in chapter 3).

$$RE_2 = (C_1, Q_{text}, \mathcal{P}_2) \tag{4.5}$$

where $\mathcal{P}_2$ refers to a keyword-based text retrieval approach.

Some examples of queries for FTS engine are:

[95] http://albali.lsi.uned.es/ accessed at 23/07/2015

videos de goles del fútbol club Barcelona (*videos of goals of football club Barcelona*)	The information need associated to this query are videos containing goals both scored by or to football teams of Barcelona (F.C. Barcelona, Espanyol, etc.).
ganador del Tour de Francia del año 2009 (*winner of the France Tour of year 2009*)	Although this query is not a question, its information need is the name of the winner of 2009 France Tour.

3. Ontology-based Search (**ObS**)[96] offers three different ways of searching inside the ontology: (a) textual search, (b) concept search and (c) SPARQL[97] search (defined in equations 4.6, 4.7 and 4.8).

   (a) Textual Search (**Textual-ObS**): uses a textual query to retrieve concepts from the ontology searching over textual metadata properties of the ontology: title and description. The query is linguistically processed by language identification and cleaning, tokenization, entities extraction using dictionaries, partition judgment and linguistic annotation. Besides, named entity recognition using own Linked Open Data (LOD)[98] dictionaries is performed. These metadata is added to a Lucene index that is requested with the query. This retrieval engine is formally defined in equation 4.6 (based on the formal model described in chapter 3).

$$RE_3 = (C_4, Q_{text}, \mathcal{P}_3) \tag{4.6}$$

   where $\mathcal{P}_3$ does keyword matching between query and textual metadata on ontology objects (title and description). It returns a set of results $\mathcal{S} = \{\mathcal{R}_1, \ldots \mathcal{R}_N\}$ where $\mathcal{M}(\mathcal{R}_n) = conc\ \forall n$.

   Some examples of queries for **Textual-ObS** engine are:

[96] `http://buscamedia.isoco.net/m3repository/index.php` accessed at 23/07/2015.

[97] SPARQL is a query language for RDF. *'SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware'.* Taken from http://www.w3.org/TR/rdf-sparql-query/.

[98] http://linkeddata.org/ accessed at 23/07/2015

videos de goles del fútbol club Barcelona (*videos of goals of football club Barcelona*)	The information need associated to this query are videos containing goals both scored by or to football teams of Barcelona (F.C. Barcelona, Espanyol, etc.).
imágenes natación (*swimming images*)	The user wants to visualize images from swimming. So no specification has been made, whatever the image of swimming would be interesting.

(b) Concept Search (**Concept-ObS**): it gets from the ontology all information (individuals, classes, etc.) related to the concept received as input. This retrieval engine is formally defined in equation 4.7 (based on the formal model described in chapter 3).

$$RE_4 = (C_4, Q, \mathcal{P}_4) \tag{4.7}$$

where:

- $\mathcal{M}(Q) = conc$
- $\mathcal{P}_4$ is a boolean matching between concept identifiers

It returns a set of results $\mathcal{S} = \{\mathcal{R}_1, \ldots \mathcal{R}_N\}$ where $\mathcal{M}(\mathcal{R}_n) = conc \; \forall n \in [1, N]$.

Some examples of queries for **Concept-ObS** engine are:

`http://www.buscamedia.es/ontologies/M3/logo/FC_Barcelona`	The query contains the ontology-concept identification of the concept F.C. Barcelona. The system should return all the information related to the football team.

(c) SPARQL Search (**SPARQL-ObS**): allows the request of SPARQL queries against the ontology. As defined in the SPARQL Query Language Specification[99],

> *SPARQL can be used to express queries across diverse data sources [...] SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also*

[99] http://www.w3.org/TR/sparql11-query/ - SPARQL 1.1 Query Language

supports aggregation, sub-queries, negation, creating values by expressions, extensible value testing, and constraining queries by source RDF graph. The results of SPARQL queries can be result sets or RDF graphs.

This search engine returns a set of results that are ontology triplets. This retrieval engine is formally defined in equation 4.8 (based on the formal model described in chapter 3).

$$RE_5 = (C_4, Q, \mathcal{P}_5) \tag{4.8}$$

where:

- $\mathcal{M}(Q) = txt$ (text must be formatted as SPARQL queries).
- $\mathcal{P}_5$ matches the SPARQL query against the ontology.

It returns a set of results $\mathcal{S} = \{\mathcal{R}_1, \ldots \mathcal{R}_N\}$ where $\mathcal{M}(\mathcal{R}_n) = trip \; \forall n \in [1, N]$. These results are the triplets contained in the ontology that answer the query.

Some examples of queries for **SPARQL-ObS** engine are:

`SELECT DISTINCT ?p ?o`
`WHERE {`
`<http://www.buscamedia.es/ont/M3`
`#LaSexta-24-10-2> ?p ?o`
`}`
	This query searches for all the information related to the concept of the ontology with identification `http://www.buscamedia.es/ont/M3#LaSexta-24-10-2`. The results will include every triplet related to the input concept.

`SELECT DISTINCT *`
`WHERE {`
`?s <http://www.w3.org/ns/ma-ont`
`#locator> ?o`
`}`
	This query searches for all the multimedia elements which contain a *locator* property (and their related locator). The results will include every multimedia object that contains a locator (information about its physical location).

4. OCR in Images (**OCRI**)[100] receives an image and retrieves the existing text on it. The result is a text (set of tokens) elements present in the image (subtitles,

[100] http://213.37.131.162:8082/buscamedia/includes/api/api-analisis.wsdl accessed at 23/07/2015.

text boxes, text on logos, etc.). First of all, the engine identifies the areas that possibly contain text. These areas are known as pills. The pills are obtained by applying the Homogeneous Texture Descriptor (HTD) [Manjunath et al., 2001] and discriminating false positives applying two classifiers based on Support vector machine techniques (SVM) [Burges, 1998]. If both classifiers deny the pill, then it is not considered as containing text. Once the pills have been identified, a sequence of tokens is generated using a free OCR software called Tesseract[101] [Smith, 2007].

$$RE_6 = (Q, \mathcal{P}_6) \tag{4.9}$$

where $\mathcal{M}(Q) = img$ and $\mathcal{P}_6$ applies pills detection and OCR (Optical Character Recognition).

An example of query for OCRI engine is shown in figure 4.4. Using this figure as query the RE_6 should return the text contained inside it: *'SALAMANCA, ESTA MAÑANA. Susto en el Helmántico por el desmayo de Miguel García (Salamanca, this morning. Huge fear in El Helmántico when Miguel García collapsed)'.*

Figure 4.4: Image query example containing the text: 'Salamanca, this morning. Huge fear in El Helmántico when Miguel García collapsed'

5. Object Detection in Images (**ODI**)[102] retrieves the existing objects in the query image and returns a set of concepts represented as terms. It uses the visual

[101] http://code.google.com/p/tesseract-ocr/ accessed at 23/07/2015

[102] http://buscamedia.bilbomatica.es:4156/ObjectAnnotation-Service.asmx?WSDL accessed at 23/07/2015

attention algorithm proposed in Itti and Koch [2000], which detects a set of specific locations over the entire image, and establishes an order in which visual attention will circulate them. After that an algorithm based on SURF [Bay et al., 2008] is applied for selection of interesting objects. The formal definition is shown in equation 4.10.

$$RE_7 = (Q, \mathcal{P}_7) \tag{4.10}$$

where $\mathcal{M}(Q) = img$ and $\mathcal{P}_7$ extracts objects in the image.

Using figure 4.4 as query, the retrieved objects should be: *football player, stadium, person, referee, ball, ground.*

6. Audio Transcription (**AT**) transcribes the incoming audio file. It returns the textual transcription together with temporal information. It uses Windows Speech Recognizer (WSR)[103] and Dragon Naturally Speaking (DNS)[104] to perform the audio transcription. The audio transcription engine is completely described in section 4.2.5.

$$RE_9 = (Q, \mathcal{P}_9) \tag{4.11}$$

where $\mathcal{M}(Q) = aud$ and $\mathcal{P}_9$ is a speech recognition algorithm.

4.2.5 Audio Transcription

Due to the fact that audio transcription retrieval engine has been developed by the author of this thesis, this component is described deeply in this section.

The functionality of a traditional information retrieval system is to provide information to solve certain user information need. An audio transcription system is different because its purpose is to transform the mode of the input (audio) in a different mode (text) but the information is not changed (it remains the same).

The first thing needed when implementing an audio transcription engine is the transcription software. This first experiments we conducted with speech recognition

[103]Using version 5.1http://msdn.microsoft.com/en-us/library/ms723627(v=vs.85).aspx.
[104]Using version 12.5.1http://www.nuance.com/dragon/index.htm.

systems are described in Schneider et al. [2009]. Three automatic transcription systems were evaluated: IBM ViaVoice, Dragon Naturally Speaking and Sail Labs' Media Mining Indexer (MMI). They are all commercial voice recognizers, and our aim was to compare them in order to choose the most appropriate one to accomplish audio-query transcription. ViaVoice and Dragon are speaker-oriented speech recognizers and they need a previous training process. However, we did not make a conventional training, but a multiuser one, using 10 different trainers, each one reading sentences from the basic text training provided by both programs. Finally, the main advantage of MMI version 5.0 is that it does not need previous training. The output is also phrase by phrase.

The scenarios where the program has been tested are two:

- **Question Answering System scenario**: we tested the recognizers using as input 163 audio files containing questions read by 10 individuals (both sexes, different ages). They were short questions, asking information about important figures, celebrities, places, dates, etc. Some examples are: ¿Qué es BMW? (*What is BMW?*), ¿Quién recibió el Premio Nobel de la Paz en 1989? (*Who did win the Nobel Peace Prize in 1989?*). The recognizers were used to convert speech to text and later to send it to the question answering system.

  The evaluation result of the recognition rate is shown in figure 4.5. All systems are performing over a 60% of correct words rate.

- **Audio-Video transcription system**:

  - **Video transcription for Information Retrieval**: this work is focused on the use of a speech recognizer for making automatic transcriptions of audio and, subsequently, retrieving information from the resulting texts. For this task, the Media Mining Indexer (MMI) was the chosen recognizer due to problems with ViaVoice to integrate audio files as input. As input, two newscasts video files were used; both of them last half an hour, and the difference between them is that while the first one is a national newscast, the 24h newscast addresses to an international audience. The results are presented in the table 4.3.

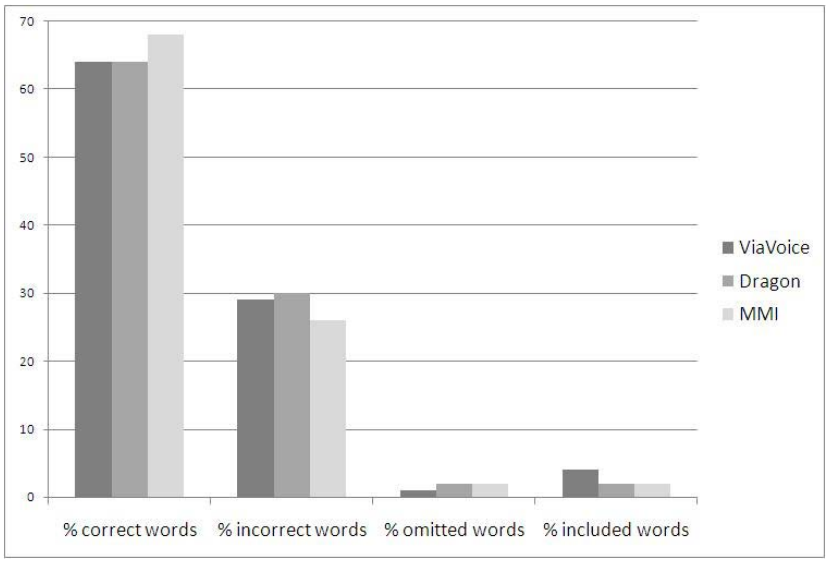


Figure 4.5: Accuracy of the three automatic speech recognizers in question answering scenario

The difference between both results relies on the audio files used for the second test, which presented a higher noise level and this is reflected in the numeric results.

– **Real-time captioning system in a classroom**: another important scenario was a subtitling application for students with hearing impairment that transcribes the teacher's speech with the help of an ASR system, converting the spoken lesson into a digital resource. These media is available in real time for deaf students in form of captioning or as plain text, in paragraphs, where

%	Newscast	Newscast 24h
Correct words	55	32
Incorrect words	32	48
Omitted words	7	9
Inserted words	3	9

Table 4.3: Percentage results of the transcription process

the user can navigate the whole transcription. A secondary task, apart from live subtitling, is the possibility of retrieving learning objects using subtitles to index video recorded in classrooms and helping students with disabilities in the learning process. The evaluation was carried out at the Carlos III University of Madrid during a 3th year subject of Computer Science degree called "Database Design". The teacher previously trained Dragon Naturally Speaking version 9 (DNS). Training duration was 30 minutes approximately, reading specific texts given by both ASR products. Additionally, specific vocabulary of "Database Design" subject was independently introduced and trained. Four experiments were performed: (1) speech recognizer's basic model, (2) basic model and training, (3) basic model and specific vocabulary and (4) basic model, training and specific vocabulary.

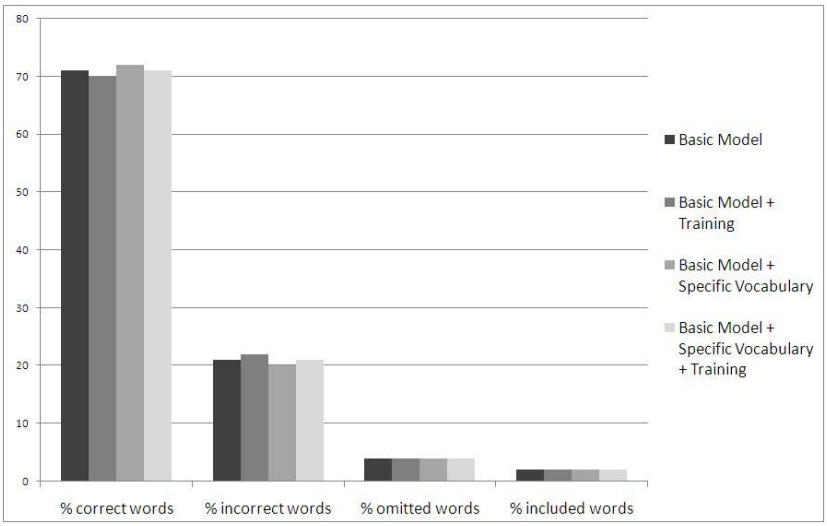


Figure 4.6: Comparison of four tests in the real-time captioning scenario

The results obtained after the comparison show a high degree of accuracy for non-structured text, although it is usually poorer as the comparison process was not designed to work with this kind of texts. The scenario for this task (a classroom) involves dealing with spontaneous speech, even though the discourse is previously planned. This means the existence of typical elements of spontaneous speech as disfluences, self-interruptions, false starts, hesitations, all of which make the recognition process difficult. Owing to

this fact, there is not much variation between the four tests, as training and vocabulary insertion do not provide better results. Moreover, keywords are not distinguished from stopwords, so, even introducing specific vocabulary, the total percentage does not improve as it is made up including stopwords.

Once we evaluated the commercial ASR software, we realized that there was no evaluation methodology for transcription systems, so we decided to create one. This methodology is described in González et al. [2013]. The final objective of the methodology is to facilitate the evaluation process of ASR products to help us to select adequate software in a particular scenario that requires voice recognition. First of all a methodological framework to design and develop tests must be defined. This methodology is composed by five steps explained next. Actually, these steps are not fully independent, there are relationships among them. For instance, the corpus preparation is influenced by the evaluation software to be used (the transcriptions of videos have to be formatted according to the required input in the evaluation system). In a similar way, the definition and selection of evaluation scenarios also affects corpus preparation. For example, if a scenario to test the performance of an ASR system with a specific speaker has to be defined, then the corpus has to contain enough video resources of this speaker.

1. The central step in the methodology is to **define and select the scenarios** that will be used for evaluating. In this case, the parameters to be considered are: domain - which takes into account whether the domain of the audio (video) is focused on a specific matter or deals with general themes; speaker - that considers if there are one or several speakers in the audio (video); training if the ASR system is going to be tested with no training, trained for a specific speaker or for several speakers; test that specifies the videos to be used in testing.

   Using these characteristics seven resulting scenarios can be defined: (a) evaluation without training; (b) evaluation with acoustic model training; (c) evaluation with previous training (language model and acoustic model); (d) evaluation with speaker-oriented training; (e) evaluation with specific vocabularies; (f) evaluation combining specific vocabulary and speaker dependence; and (g) evaluation without language model.

2. **What will be measured?**: To evaluate speech recognition systems, the output of the ASR system, called hypothesis text, is compared to a literal transcription of input audio, denoted as reference text. Standard measures used in speech recognition evaluation are [Bernsen et al., 2007]:

   - Word Error Rate: it measures the percentage of incorrect words (p_s-substitutions, p_i- insertions, p_b-eliminations) regarding the total number of words.

     $$WER = \frac{n_e}{p_t} = \frac{p_s + p_i + p_b}{p_t} \tag{4.12}$$

     where n_e is the total number of errors in hypothesis text and p_t is the number of total words in the reference text.

   - Word Accuracy: it measures the total number of correct words regarding the total number of words.

     $$W_{Acc} = 1 - WER = \frac{p_c}{p_t} \tag{4.13}$$

     where p_c is the total number of correct words in hypothesis text.

3. After defining the measurements that are going to be used to evaluate the system, next step is to **select the evaluation software** to test the quality of recognition process. A well-known software to evaluate speech recognition is Sclite[105] that is part of the Scoring Toolkit developed (SCTK) developed by NIST (National Institute of Standards and Technologies). The goal of Sclite is to evaluate an ASR system by comparing a manual transcription with the automatic transcription obtained from the ASR.

4. **Create and prepare a Corpus**: the first collection is composed of 15 generic TV Broadcast news (with duration of one hour each), 10 videos about sport news videos and 10 videos containing weather forecasts (approx., 10 minutes each). These resources had to be split in segments of approx. 10 minutes due to (a) allowing configuring different training-testing parts; and (b) software limitations both in ASR system and in NIST Score Toolkit evaluation software. Then, these videos are classified according to different parameters: audio format, domain, speakers, noise, music and other characteristics that should have correspondence with the scenarios defined in the second step.

[105] `ftp://jaguar.ncsl.nist.gov/current_docs/sctk/doc/sclite.htm` accessed at 23/07/2015

5. The last step is to **prepare evaluation environment and run evaluation** on the DNS software, so three scenarios described in step 1 have been selected (a, c and d). DNS provides two manners to train a speaker model, one is using the commercial version and other is using different functions that are provided by Dragon SDK. Four different trainings were defined: (i) evaluation without training: using the default acoustic and language model provide by DNS (scenario a); (ii) evaluation with previous speaker independent training (scenario c); (iii) evaluation with specific vocabularies (scenario e); and (iv) evaluation combining specific vocabulary and speaker dependent training (scenario f).

   Table 4.4 shows the experiments that have been completely developed and evaluated. Word accuracy values are very similar in the three cases. We believe that training using video segments where 10/12 different speakers are taking part, with noise, music and overlapping voices is not a good material to train user models.

%	Scenario (a)	Scenario (c)	
	Without Enrollment	Short Enrollment	Long Enrollment
Correct	68,8%	69,8%	71,7%
Substitutions	15,8%	14,3%	13,8%
Deletions	15,4%	15,9%	14,5%
Insertions	3,8%	3,3%	3,9%
Word Accuracy	64,9%	66,5%	67,8%

Table 4.4: Preliminary results using DNS system.

First accuracy figures shown in table 4.4 should be taken as preliminary results, showing an almost negligible accuracy increase comparing trained and no trained experiments.

After defining the methodology, we tried to apply the transcription engine to solve a real problem. We have made a named entities correction proof of concept in Schneider et al. [2014]. This proof of concept corrects errors in the recognition of named entities in queries.

ASRs are not able to recognize entities that are not present in its vocabulary so the problem considered in Schneider et al. [2014] is the misrecognition of named entities in Spanish voice queries. Most works on this area try to modify the acoustic or language models of the ASR, but sometimes there is no possibility of making any change in the ASR system, e.g. if a real-time reaction is needed so there is no time to modify the acoustic model or if some predefined system (as Android or IPhone Speech Recognition) is integrated into an application. In this case, the problem has been addressed from that point of view: there is no possibility of making any change in the ASR system.

As can be seen in the examples of table 4.5, the main problem lies in the entities that are falsely recognized, i.e. the obtained entity is not the one that was said ('Woody Allen' - 'Raúl González'), or it is not even a named entity, i.e. getting a common noun when a named entity was said ('Kun Agüero' - 'unahuelga').

Original Query	Recognized Query
¿Cuál fue la última película dirigida por Woody Allen? (What was the last film directed by Woody Allen?)	¿Cuál fue la última película dirigida por Raúl González? (What was the last film directed by Raúl González?)
¿En qué equipo juega Kun Agüero?(Which team does Kun Agüeroplay in?)	¿En qué equipo juega una huelga? (Which team does unahuelga play in?)

Table 4.5: Examples of misrecognized Named Entities

The objective is to provide alternative entities to those incorrectly recognized or misrecognized by retrieving phonetically similar entities. This system is domain-dependent, using sports news, specifically football news, regardless of the automatic speech recognition system used. The correction process exploits the query structure and the semantic types of phrases to detect where a named entity appears (for instance, the query "Which team does Cristiano Ronaldo play for?" has the structure "which team does ##FOOTBALL PLAYER play for?" where the semantic type ##FOOTBALL PLAYER points a named entity susceptible of being reviewed. The detection of a misrecognized named entity is done by searching it in the previously defined dictionaries (if the dictionary does not contain the named entity then it is considered to be incorrectly recognized).

The treatment needed on these entities is essentially a correction assuming that in some cases the entity will not be correctly recognized or even is not an entity (see the previous example of "Football player").

As the main difference with the related work, it can be pointed out that this proposal is a dictionary-based system that works directly over named entities instead of trying to correct each word or the whole transcription. Considering that there is no specific work on named entity correction in Spanish voice queries the objective is to perform this correction through a post processing over transcribed queries with ASR-independence (considering that it is not possible to modify nor the ASR nor its models). The domain is limited to sports news to get the named entities dictionary.

The system must find the most suitable alternative to the entities received inside the input query. To search for these alternatives a phonetic comparison between the recognized entity (by the ASR) and the entities stored in the dictionary is used and the highest scored entity is obtained (by using string comparison measurements). This functionality (together with the system's architecture) is shown in figure 4.7.

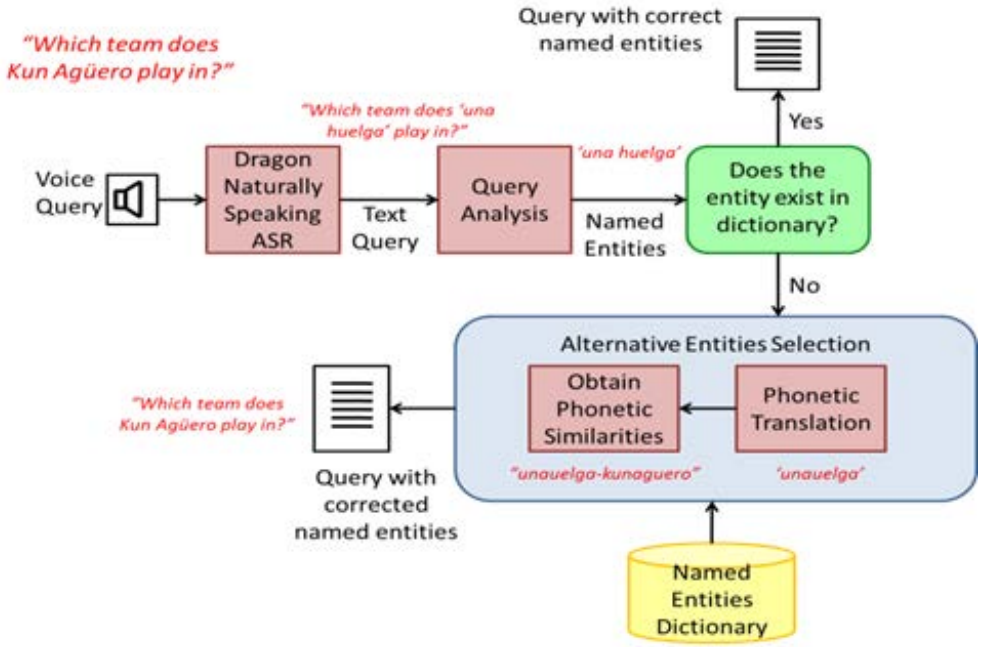


Figure 4.7: Entity correction proof-of-concept architecture from Schneider et al. [2014]

The functionality of the system is structured in three main parts. Firstly, the ASR transcribes the input voice query providing a textual query. In this case two different ASR systems have been used: Dragon Naturally Speaking and Windows Speech Recognizer.

The second part of the architecture is composed by the entity extraction module. It takes care of the query analysis and searches entities inside it. This search is performed by means of a rule-based system that considers five different query patterns. These patterns are shown in table 4.6.

Query Patterns
¿En qué equipo juega ##JUGADOR? (What team does ##PLAYER play for?)
¿Quién marcó el último gol en el estadio ##ESTADIO? (Who scored the last goal in the stadium ##STADIUM?)
¿Quién es el máximo goleador del ##EQUIPO? (Who was the maximum scorer of ##TEAM?)
¿Cuántos goles ha marcado ##JUGADOR este año? (How much goals has ##PLAYER scored this year?)
¿Cuántos penaltis se pitaron en el último partido que se jugó en ##ESTADIO? (How many kick goals were dictated in the last game played in ##STADIUM?)

Table 4.6: Available Query Patterns

To determine the corresponding pattern for the input query a direct comparison is not appropriate because it can contain transcription errors. Due to that, a bag of words approach is used. It counts the number of words of each pattern contained in the input query. Once the pattern has been determined, the entity is extracted by means of its position in the query. The next step of the system is checking whether the extracted entity has been correctly recognized or not. This functionality is performed by determining the presence of the entity in the dictionaries. If so, the entity is considered to be properly recognized and no correction is done.

Char.	Phon.	Char.	Phon.	Char.	Phon.	Char.	Phon.	Char.	Phon.
a	a	f	f	k	k	o	o	t	t
b	b	g	g/j	l	l	p	p	u	u
c	c	i	i	m	m	r	r/R	v	b
d	d	j	j	n	n	s	s	w	ui
e	e	y	i	ñ	N	z	z	x	ks

Table 4.7: Spanish phonetic letter correspondence between characters (Char.) and phonemes (Phon.)

On the contrary, if the entity does not appear in the dictionaries, then alternative entities for that incorrectly recognized (or misrecognized) entity are provided. A phonetic representation of the input entity is generated using a rule-based system implemented as an adaptation of the work of Gil [2007] and LivingSpanish [2011] for Spanish phonetic letter correspondence. This representation is shown in table 4.7.

The similarity between the phonetic representation of the recognized entity and the phonetic representation of the named entities of the dictionary is evaluated. Indeed, several measures have been tested, such as Euclidean, Monge-Elkan, Levenshtein, Needlemann-Wunsch, Smith-Waterman, Gotoh or Smith-Waterman-Gotoh, Jaro, Jaro-Winkler and Soundex distances. A complete description of these measurements can be found in Gusfield [1997] and Cohen et al. [2003].

The implemented dictionary is composed by a set of named entities together with its associated information (in XML format). It structure can be seen in table B.1 (see annex B).

The dictionary is composed by a set of properties and 2004 different named entities. The properties are total number of stored entities, total number of each type of entity (players, stadiums and teams) and the times that each type of entity has been selected as a suitable alternative. The entities are divided into 1874 football player names, 42 stadium names and 88 team names. Besides, each entity is composed by its associated text, the type it belongs to, a popularity score defined by an expert and the number of times it has been selected as alternative.

The first evaluation was carried out using 168 Spanish voice queries read by 7 different users. These queries are uniformly distributed over the five query patterns. Some examples are shown in table 4.8.

Original Query	Transcribed Query (with one ASR)
¿En qué equipo juega juan-josécollantes?	El equipo, Juan José Collantes
¿Quién marcó el último gol en el estadio los pajaritos?	Quien marcó lo temor en el estadio los pajaritos
¿Quién es el máximo goleador del valencia?	Quién es el máximo goleador del Valencia

Table 4.8: Examples of input queries read by users

The queries have been transcribed using both ASRs. The first ASR was used with four different acoustic models trained with videos of different length. The first model (DNS-1) was not trained; the second (DNS-2) was trained only with approx. 5 minutes of sport news videos; the third model (DNS-3) was trained with 50 minutes of sport news videos; and the fourth (DNS-4) was trained with 40 minutes of football news videos. The second ASR (WSR) was not trained and only its default model was used.

The first test is performed to validate the functionality and performance of the entity classification module. In order to do that, the entities were manually extracted from the transcribed queries and then classified into types for using them as reference. The five different ASR models are tested and four different classification techniques are shown. The first technique is a direct comparison between the transcribed query and the patterns; the second is a bag-of-words technique (complete BoW) that uses all the words (including the entity tags (#footballplayer)); the third improves the bag-of-word technique by eliminating the tags (Limited BoW); and the last performs a phonetic comparison between the query and the patterns.

The results obtained by the entity type classifier and the entity extractor are shown on the next table (table 4.9). As can be seen, the phonetic comparison classification is the best approach.

	WSR	DNS-1	DNS-2	DNS-3	DNS-4
Direct Comparison	0	0	0	0	0
Complete BoW	78,57% (132)	70,83% (119)	72,62% (122)	73,21% (123)	58,33% (98)
Limited BoW	82,74% (139)	64,88% (109)	72,02% (121)	70,83% (119)	54,17% (91)
Phonetic Comparison	88,69% (149)	86,9% (146)	90,48% (152)	89,88% (151)	77,38% (130)

Table 4.9: Results of Entity Classification Module Validation using five speech recognition models (four using Dragon Naturally Speaking (DNS) and one using Windows Speech Recognizer (WSR)) and four different classification techniques: direct comparison, bag-of-words technique using every word, bag-of-word technique eliminating the tags and phonetic comparison. In brackets it is shown the number of entities.

The second test was performed to validate the phonetic representation system. The phonetic representation that was finally used works properly as long as the entities are 'Spanish' entities while it fails with entities from other languages.

Cristiano Ronaldo (kristianoronaldo) and Lionel Messi (lionelmessi) are well represented as expected and Schweinsteiger (scuieinsteijer) is not. Different is the case of Hamit Altintop (amitaltintop) that was correctly represented although it was not expected.

The corpus used for that task was composed by 168 entities. These entities were introduced into the phonetic representation system and the output of each entity was manually revised to determine if it was correctly represented. The amount of correct represented entities was 150 entities. That leads to an accuracy of 89,29

It can be remarked that all the entities that were Spanish names were correctly represented while the errors occur when there is a foreign entity (giorgioventurin-jiorjiobenturin, ilijanajdoski-ilijanajdoski).This is a known problem of the phonetic representation module since it has been only implemented for entities in Spanish.

The last test validates the module that retrieves alternative entities using the same corpus of 168 entities. For this purpose some different phonetic distance measurements where used. The best comparison measurements for phonetic comparison are

Levenshtein Distance and Monge-Elkan-Levenshtein Distance obtaining figures near to 56.55% in Top@10 (Levenshtein) and 50.60% in Top@1 (Monge-Elkan-Levenshtein).

The phonetic entity correction system in-creases the accuracy in both cases: using WSR it increases 19,65% and with DNS the increment is a 19,05% for the total amount of entities.

The results of the entity correction module using a system with multiple dictionaries, i.e. it uses a different dictionary for each entity type. It depends on the performance of the entity type extraction but increases the entity correction in 3% approximately.

These are promising results considering that it was a proof-of-concept.

The correction of named entities in IR systems accessed by voice is absolutely necessary. There are mainly two reasons for that; on the one hand the entities are an essential unit of information for IR systems, on the other hand in most scenarios the acoustic and language model of the ASR cannot be modified to improve the results, letting this to a post processing after the recognition process.

Some comparison distance measurements were tested and finally only two of them were selected for final tests. These measures have proved very useful when making phonetic comparison. Additionally, the results are even improved when the arithmetic mean between both measures is used as a new measure (Monge-Elkan-Levenshtein).

After a preliminary evaluation, the 52% (approx.) of entity alternatives are right choices, and after making a qualitative assessment, it can be said that whenever the entity has not been recognized, the system will be able to offer an appropriate alternative.

This work has got promising results but is still in an early development stage. There are some improvements that can be outcome to the system. The first improvement would be the adaptation of the phonetic representation system to take into account different pronunciations (accents) and especially words in other languages. Besides, some ASRs return acronyms and the phonetic expansion of these acronyms could be useful for the desired purpose.

Every experiment previously made has helped us determine the best ASR system to use as transcription engine. For this prototype the Windows Speech Recognizer (WSR)[106] has been used as transcription software. It has been selected because of its

[106]http://msdn.microsoft.com/en-us/library/jj127860.aspx accessed at 23/07/2015

good transcription rate, its easiness to be embedded and the easiness to train it or to use it without training.

4.2.6 Orchestrating retrieval engines (Handler)

As it has been previously described, the prototype uses several retrieval engines to get the information that is returned to the user. A handler is required to decide which retrieval engines will be requested with each query (see section 3.2.4).

Since the model has been particularized for rule-based handler, the handler implemented for this prototype is also based on rules. Each of the rules (whose structure is shown in equation 4.14) consists of a number of conditions (left side of the assignment) and a subset ($\mathcal{E}'$) of the available retrieval engines ($\mathcal{E}$) (right side of the assignment).

$$Conditions \ \rightarrow \ \mathcal{E}' \tag{4.14}$$

The implemented rules in the prototype use two types of information in the conditions: mode ($\mathcal{M}(Q)$) and type ($\Psi(Q)$) of the query, which values are shown in table 4.10. More modes and types can be added but the prototype is limited to these.

$$\mathcal{M}(Q) = value \ and \ \Psi(Q) = value \rightarrow \mathcal{E}'\{RE_1, \dots RE_h\} \tag{4.15}$$

where $\{RE_1, \dots RE_h\}$ is the set of REs that are requested.

Two different handlers have been implemented in the prototype.

1. The first handler (considered as 'baseline') requests every available REs (just considering input query mode limitations). It is a single rule strategy and is represented as

   $$\mathcal{H}_1 = (\mathcal{E}, Q, \Xi_1) \tag{4.16}$$

   where

   $$\Xi_1\{* \rightarrow \mathcal{E}' = \{RE_1, \dots, RE_N\} \ \} \tag{4.17}$$

   where N is the number of REs.

2. The second handler is an "'heuristic rule-based strategy"' supported by predefined rules.

   $$\mathcal{H}_2 = (\mathcal{E}, Q, \Xi_2) \tag{4.18}$$

Property	Value	Description
$\mathcal{M}(Q)$	txt	text query
	aud	audio file
	vid	video file
	img	image (file or content)
	conc	semantic concept (textual identifier)
	trip	semantic triplet (rdf format)
	inst	semantic concept instance identifier
$\Psi(Q)$	*	every query
	question	a complete question
	short	textual query with three or less tokens
	long	textual query with more than three tokens
	voice	the query is an audio file
	multi	queries combining text and image

Table 4.10: Possible values of $\mathcal{M}(Q)$ and $\Psi(Q)$

where each rule defined in Ξ_2 uses mode ($\mathcal{M}(Q)$) and type ($\Psi(Q)$) of the query as conditions. The following rules have been defined by the functionality of the retrieval engines (QAS needs a question at the input, so requesting it with other type of query would be useless) and the expected answer (a question needs a concrete answer while a short query could accept results from every type):

(a) Only text as query ($\mathcal{M}(Q) = txt$): three rules are defined depending on the query type which could be *question* (a linguistically complete question), *short* (three or less tokens) or *long* (more than three tokens).

$$\Psi(Q) = question \rightarrow \{RE_1, RE_2\} = \{QAS, FTS\} \tag{4.19}$$

$$\Psi(Q) = long \rightarrow \{RE_2\} = \{FTS\} \tag{4.20}$$

$$\Psi(Q) = short \rightarrow \{RE_2, RE_3\} = \{FTS, ObS\} \tag{4.21}$$

(b) Voice query ($\mathcal{M}(Q) = aud$): the query file is transcribed using an automatic transcription service and then it is treated as a textual query.

$$\Psi(Q) = voice \rightarrow \{RE_9\} = \{AT\} \tag{4.22}$$

where RE_9 is an automatic audio transcription retrieval engine (see section 4.2.5).

(c) Multi query ($\Psi(Q) = multi$): the query is divided into two parts: text and image. The text is used as an independent textual query while the image is analyzed by the image REs (OCRI and ODI) obtaining text that later is also managed as a textual query.

$$\begin{aligned} &\mathcal{M}(Q) = txt \rightarrow \{equations 4.19 - 4.22\} \\ &\mathcal{M}(Q) = img \rightarrow \{RE_6, RE_7\} = \{OCRI, ODI\} \end{aligned} \tag{4.23}$$

The final set of rules of the second handler is shown in equation 4.24.

$$\Xi_2 = \left\{ \begin{array}{rl} \mathcal{M}(Q) = text \ and \ \Psi(Q) = question & \rightarrow \mathcal{E}' = \{QAS, FTS\} \\ \mathcal{M}(Q) = text \ and \ \Psi(Q) = long & \rightarrow \mathcal{E}' = \{FTS\} \\ \mathcal{M}(Q) = text \ and \ \Psi(Q) = short & \rightarrow \mathcal{E}' = \{FTS, ObS\} \\ \mathcal{M}(Q) = audio \ and \ \Psi(Q) = voice & \rightarrow \mathcal{E}' = \{AT\} \\ \mathcal{M}(Q) = image \ and \ \Psi(Q) = multi & \rightarrow \mathcal{E}' = \{OCRI, ODI\} \end{array} \right\} \tag{4.24}$$

4.2.7 Heterogeneous results management: fusion of results

Using multiple retrieval engines requires implementing a results' fusion module (as explained in section 3.2.5). This module receives all the results obtained from each retrieval engine and manages to get a single homogeneous set of results. Besides joining the results in the right order, it has to convert the structure of the results of each RE to a 'unified' format. A unified format does not mean that every result has the same information, but it follows a common structure, although some results may not have all fields completed. A textual document contains title, content and a snippet of sample while an image will only contain title and content.

Only one fusion module has been implemented and it is based on a Round Robin strategy [Silberschatz et al., 2008]. This functionality is depicted in figure 4.8. It consists on adding the first result of the first resuts' set, then the first result of the second results' set, and so on until every results' set has been used. Then, the second element of every results' set is taken, and so on until every results' set is empty.

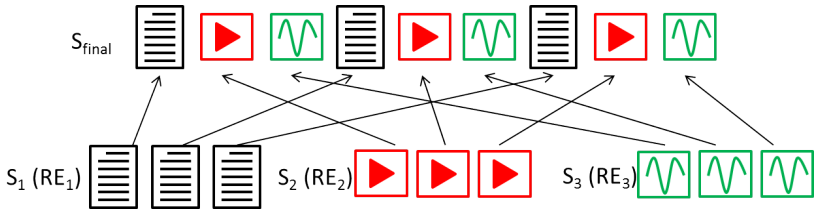


Figure 4.8: Round Robin algorithm example combining three results' sets (texts, audios and videos)

The formal definition of the round robin strategy is displayed in equation 4.25, which determines the final position of the j^th result of the i^th retrieval engine ($d_{i,j}$).

$$rank(d_{i,j}) = (N_E \cdot j + i) - N_E \tag{4.25}$$

where N_E is the number of results' sets that are combined.

The fusion is the previously presented by the IMIR system, which returns an ordered list of results. Besides the fusion of results in a single list, the prototype uses other ways of results' visualizations in the GUI (deeply explained in section 4.2.8):

1. Visual Fusion: it is achieved by showing the results (title) in visual groups, such as results' lists of a particular type or word clouds (see figure 4.15).

2. Semantic Fusion: it is performed to pool the results (in this case only the ontology concepts) based on their semantic categories (see figure 4.14).

4.2.8 Graphical User Interface

Once the multimodal search system was developed, we needed a graphical user interface (GUI) to make it usable by users.

Besides the ease of use, the implementation of this interface allows registering user interactions. Because of the need of registering some user actions that were directly

related with visualization purposes, the graphical interface is responsible for the logging process.

Figure 4.9 displays a screen shot of the interface with the available search modalities.

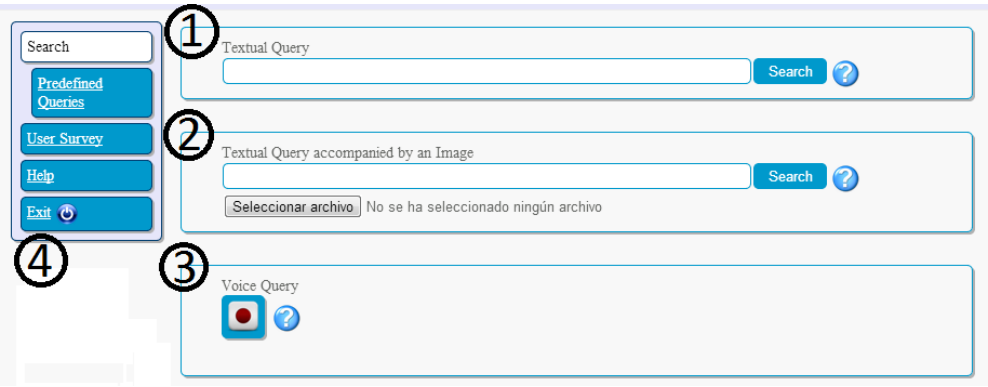


Figure 4.9: Screen shot of the query boxes implemented in the prototype

There are four clearly defined parts marked with numbers. (1) represents the textual query box; (2) marks the textual and image query box; (3) shows the voice query box; and (4) denotes the lateral navigation menu, that allows the navigation through the different graphical interfaces.

Meanwhile, figure 4.10 shows the list of results for the textual query *'Barcelona'*. The list contains two types of results: semantic concepts and news documents.

Figure 4.10: Screen shot of the prototype showing the results list for textual query *'Barcelona'* taken from Arguello et al. [2012]

The available or related multimodal content (videos, images, texts or audios) appears at the bottom of each result (remarked with red box at the bottom of the figure). The relevance feedback is made by the three-color faces (green, orange and red) each symbolizing a relevance value (remarked with red box at the top of the figure).

The graphical user interface offers five visualization modes (explained deeply in section 5.3): individual result, lists containing results in several modes, semantic clusters, mode-specific-result lists and word clouds.

When an individual result is displayed, it shows not only its title but also its content. Figure 4.11 shows a result for query *'Barcelona'* where the document contains title, text and a video (displayed by an embedded player).

Figure 4.11: Screen shot of an individual result containing a video element and its associated text transcription.

The list containing multimodal results (results in several modes) is shown in figure 4.12. This list contains **Concepts** (from *ObS* engine) and **news** (from *FTS* engine) for the query *'Barcelona'*.

An example of mode-specific-results' list is depicted in figure 4.13. In this case, the results are only **answers** retrieved from the *QaS* engine. The requested query was *'Quién es el presidente de la UEFA? (Who is the president of UEFA?)'*.

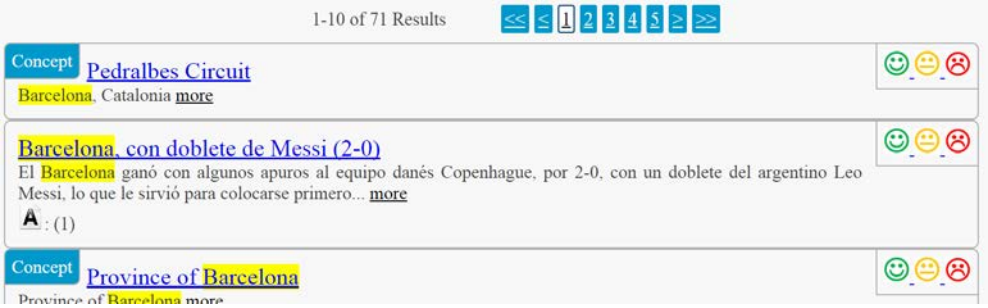


Figure 4.12: Screen shot of the combined result list containing concepts and documents.

Figure 4.13: Screen shot of the list of answers.

An example of semantic clusters is shown in figure 4.14. In this case, the results are only **semantic concepts** from the ontology (see section 4.2.2) retrieved from the *Ontology-based search (ObS)* engine. The requested query was *'Barcelona'*. The clusters are defined on the basis of the semantic groups and the hierarchy of the ontology.

An example of terms cloud is depicted in figure 4.15. The results are **answers** offered by *QaS*. The requested query was *'¿Quién es el presidente de la UEFA? (Who is the president of UEFA?)'*. As can be seen, the size of every term (see equation 4.26) depends on the relevance of the entity for the query. Although we do not retrieve the relevance from the search engine, we use the inverse of the ranking position as the relevance score.

$$size_t \propto \frac{1}{rank_t} \tag{4.26}$$

Figure 4.14: Screen shot of the semantic grouping of concepts [Gerl et al., 2012].

Figure 4.15: Screen shot of the cloud of concepts [Halvey and Keane, 2007].

The last goal of the interface is to allow the accomplishment of an evaluation (see chapter 5). This evaluation needed to store all interactions that users made on the system, and each interaction needed to have an associated user ID. For this reason, the interface offered users two ways to use it: (i) users registered by creating an account and access the system with this ID, or (ii) users accessed as anonymous users (see figure 4.16).

In addition to the initial registration, the prototype offered users the option of filling a final survey (see figure 4.16) for knowing their impressions. This form is explained in

detail in section 5.4.

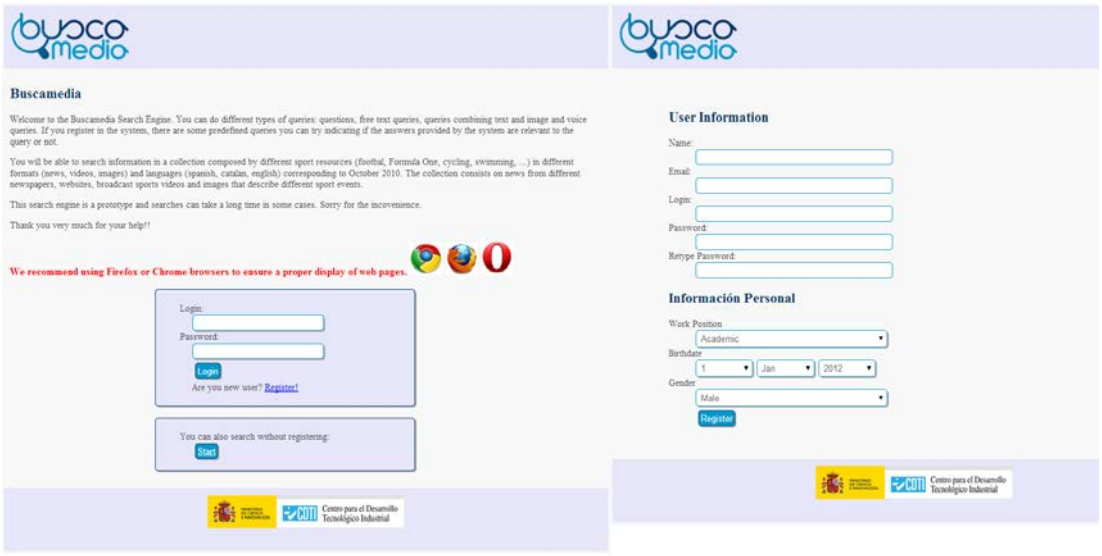


Figure 4.16: Screen shot of the access and register sites.

At this point, there is a functional multimodal IR prototype that works with six retrieval engines (several modes) and that accepts three types of queries. It also implements a handler which is based on predefined rules to manage the engines. Once the description of the prototype has been accomplished, next step is to evaluate it. Chapter 5 explains the evaluation process followed to validate and evaluate the prototype functionality and performance.

5

Analysis of the prototype functionality

After defining the model (chapter 3) and implementing a prototype based on it (chapter 4), the next step is to describe the experiments carried out to validate the prototype. The prototype is composed of six retrieval engines (full text, question answering, ontology-based, object detection in image, text detection in image and audio transcription) and it accepts tree types of queries (text, audio and combination of text and image). The prototype includes a rule-based handler whose rules have been defined manually and a results' fusion round robin strategy. The last property of the prototype is that it records every action of the users (interactions). First of all, the information retrieval accuracy of the prototype is validated. Then, this prototype is evaluated by users with the goal of recruiting user interactions. The perception of the user is analyzed using a final survey that could be filled up by users after finishing the evaluation process.

The evaluation of IR systems is usually done following two approaches: the traditional algorithmic approach (Cranfield-based experiments) and the cognitive approach (focus on cognitive structures of the user) [Olvera Lobo, 1999]. The algorithmic approach (Cranfield-based) evaluates the performance of indexing systems and users are not taken into account. In this thesis, users are involved in the evaluation by means of their interactions. Interactions are the actions the user does when (s)he is using

the system. When users are involved, Cranfield experiments are not applicable, so two different approaches appear:

- When talking about interactive IR, most of current state-of-the-art systems are evaluated following **task-oriented** methods [Kelly, 2007] where users have to complete predefined searching tasks. The goal of this evaluations is that users complete as many tasks as possible and to obtain quality measurements such as time or resources used. Usually these tasks have a concrete information need. For example, a concrete task could be that a user must search information about a good neighborhood in a city to buy a house, and (s)he wants to find relevant information to discover the best part of town depending on geographic location (proximity to schools, parks, shopping centers, etc.), public transport possibilities, etc. The relevant results have been previously defined and the evaluation consists of determining how many relevant results the user can find.

- On the contrary, in **user-centered** evaluations the main goal is to obtain users perception about the tasks they complete, without measuring time or resources used. The evaluation criteria are normally quality of the task result or the user's experience and satisfaction.[107]

In our case, we can not apply any of these methods. The task-oriented evaluation forces users to perform a defined set of queries for finding certain information. We do not want users to find concrete information, but using the system with no constraint to analyze if the prototype (together with the implemented visualizations) is useful for making multimodal queries and displaying multimodal results. It is also analyzed if the retrieved information is relevant to the query through the relevance feedback interactions.

We are following the user-centered evaluation, where the tasks we offer to the users are only using the system freely without any constrain during the evaluation process. However, because of the out-of-date collection several predefined queries were provided to users as suggestions (see Table 5.1) in order to help in formulating queries. All documents were collected in October 2010. This forced users to search information in

[107]Taken from http://www.promise-noe.eu/documents/10156/0b385617-b7f5-4aae-a108-d54f0c7d8dbb at 23/07/2015

this temporal period. In this sense, the coach of FC Barcelona in 2010 was Guardiola, while in 2012 (when the evaluation was performed) it was Tito Vilanova. Because of that, predefined queries were suggestions offered as a help for users, instead of a mandatory set of searches they had to accomplish.

Id	Query
1	¿Cuántos kilómetros recorrió Samuel Sánchez en la prueba de ciclismo de los Juegos Olímpicos? (How many miles did Samuel Sanchez travel in the cycling event of the Olympic Games?)
2	Información sobre el accidente de la foto (Information about the accident in the image) together with and image as shown next: 
3	videos goles Barcelona (Barcelona goals videos)
4	¿Quién es el presidente de la UEFA? (Who is the president of UEFA?)

Table 5.1: Predefined queries offered to the user to facilitate finding information in a period of time

The evaluation process took **2 months** (April and May 2013) and finally **233 users** participated in it. As it is said in section 4.2.8, the prototype allowed registered users. In this evaluation there was also the possibility to use the system as 'anonymous user', i.e. the system assigned an identification to the user which was not associated with its personal information, just to keep a trace of the session the user was doing. Only 27,47% of the total number of participants were registered in the system, while 72,53% preferred to use the system as an anonymous user without providing personal information.

5.1 Analysis of Queries

A total number of 981 queries was done and 239 correspond to predefined queries. 75, 64% of the queries were generated by users. Each user had made an average of 4, 58 (max. 37 and min. 1) searches per session.

%	Predefined	Self-defined	
Registered	15, 05	39, 89	54, 94
Anonymous	6, 33	38, 73	45, 06
	21, 37	78, 63	

Table 5.2: Percentage distribution of self-defined and predefined queries used by registered and anonymous users

Looking at the distribution of predefined queries (see table 5.1): queries number 1 (39, 75%) and number 4 (42, 68%) were the most executed queries. Query 3 was less used (16, 32%) while the query combining text and image (query number 2) was barely used (only 1, 6% of predefined queries). Table 5.2 shows the distribution of queries organized by user type.

Table 5.3 shows the percentages of use of each query mode over all searches made to the system.

	Text	Text + Image	Audio
Percentage	86, 5	3, 6	9, 9

Table 5.3: Percentage distribution of query modes that have been used during the evaluation process

Although text is the most intuitive way of searching information (based on current IR systems), we expected to get more users trying voice and combined (text + image) queries. Only 9, 9% of searches used voice queries and it was even lower for combined queries (3, 6%). The majority of the searches used textual queries (72, 4%).

The textual queries are classified into 4 types: question (Who is the president of UEFA?), short text with three or less tokens (videos goal Barcelona), long text with

more than three tokens (Number of football world championships of Brazil) or ontology concepts (`http://www.buscamedia.es/ontologies/M3/logo/FC_Barcelona`). Ontology concept queries are not introduced directly by users, but they are used for exploratory search once an ontology result has been retrieved by another search. The textual query variants distribution associated to every query mode is shown in table 5.4.

%	Question	Short Text	Long Text	Concept
Txt	35, 70	37, 72	10, 25	16, 33
Txt+Img	45, 45	27, 27	27, 27	0, 00
Audio	2, 22	32, 22	65, 56	0, 00

Table 5.4: Percentage distribution of text types (question, short, long or concept) classification for each query mode

The first thing that stands out is that concepts were only used as query when a text search was performed. This is consistent, since no user would speak directly the URI of a concept of the ontology or joint it to an image. When users gave a text query most of them were either full questions (35, 70%) or short texts (37, 72%), while long text queries (10, 25%) or concepts queries (16, 33%) were used far less. For the case of voice query, data inverted and the most used queries were long text (65, 56%), while short text were used less (32, 22%) and virtually no whole questions were given (2, 22%). Doing a manual analysis of the transcribed queries showed that in many cases the pronounced query was a question but the query was classified as long text. Most ASR products do not return punctuation marks and the rule that identifies a query as a question is based on two things: question marks (?) and the terms: *how, when, why, which, what and where*. If any of these circumstances is met, the query is considered as a question. Therefore, many transcribed queries were badly classified. When talking about combined query (text and image), the most used text type was questions (45, 45%), while short and long text queries were equally distributed (27, 27%). This result is also consistent, since in most cases the user wanted to get additional information to the provided image, so that the most intuitive way was a question that inquires information about the image.

5.2 Analysis of Information Retrieval Performance

We wanted to make an IR assessment such as those performed in CLEF (Cross Language Evaluation forum), in fact, we have previously carried out several works using this type of evaluation (Vicente-Díez et al. [2009], Martínez-González et al. [2009] or Pablo-Sánchez et al. [2008]) and we are familiar with it.

In this works we developed and evaluated a question answering system. It requested a pure information retrieval system (Lucene) retrieving a list of documents. Then, these documents were filtered, depending on the information obtained from the linguistic analysis of the query, and the relevant information to answer the question was extracted. While performing these works we participated in two evaluation forums (CLEF2009[108] and CLEF2010[109]) and worked in three different domains: general news obtained from EFE, wikipedia documents and a collection of legal documents (jrcacquis)[110]. A deep description of some collections is given in section 2.1. The main problem is that the CLEF collections are neither multimedia enough (they are mainly monomodal or bimodal) nor contain semantic information.

Because of that we have used the multimedia collection defined in the Buscamedia project[111] (see section 4.2.1).

A goldstandard is an element used for evaluating IR systems. It is composed of a set of queries and the corresponding relevant documents (of the available collections of documents) to each query. We did not have a gold standard from our collections of documents and creating a goldstandard for a large collection of documents to be used in evaluation tasks is extremely expensive, since a relevance judgment to each document for each query has to be assigned. Therefore, we have adopted a compromise solution: we have created a ***SilverStandard*** to evaluate our system. This silver standard was created from a set of queries posed by 233 users during the prototype evaluation process. A silverstandard follows the same concept as goldstandard with two differences:

1. The relevance judgments are assigned after requesting the retrieval engine using the query.

[108] http://www.clef-campaign.org/2009.html accessed at 23/07/2015
[109] http://www.clef-initiative.eu/edition/clef2010/working-notes accessed at 23/07/2015
[110] http://ipsc.jrc.ec.europa.eu/index.php?id=198 accessed at 23/07/2015
[111] http://www.cenitbuscamedia.es/ accessed at 23/07/2015

2. Not all documents in the collection are assigned a relevance judgment. In fact, not even all the documents returned by the retrieval system are. Relevance judgments are given only to the N first results (being typically $N \in \{5, 10, 20, 50, ...\}$).

The generation of our silverstandard is conditioned by the information we know about the queries and the results. We logged the information, encompassing 518 queries and 12945 user interactions, of the retrieval process during the prototype evaluation (eight weeks). This information contains the requested queries, the list of returned results and the rated results (as *relevant*, *irrelevant* and *neutral*). Since we had a large number of queries, the top 30 results of each query were used to make a manual rating. The queries sent to the prototype together with these assessments defined the silverstandard corpus to be used in evaluation tasks.

Once the silver standard was defined, we could evaluate the retrieval systems. In this evaluation only three (**FTS**, **QAS** and **ObS**) of the six available retrieval engines have been used. The other engines have not been used because they does not retrieve information from the collections so it is impossible to determine the relevance of the results in the silver standard for these engines. We compare four configurations of the prototype: (1) using a single retrieval engine (**QAS**); (2) using a single retrieval engine (**FTS**); (3) using a single retrieval engine (**ObS**); and (4) using every retrieval engine (**QAS**, **FTS** and **ObS**).

The most common measures for evaluating IR systems (according to the article of Kelly and Sugimoto [2013]) are *Precision*, *Recall* and *F-Measure*, but these measures have the problem that they do not consider the order of the results in their computation, i.e., if our system returns two relevant results the same precision will be obtained regardless their position: first and second or fourth and fifth. We need to consider the position of the relevant results, so this criteria has been used for selecting the measures to evaluate our prototype.

1. **R-Precision (R_p):** is defined as the precision at position R. It measures the precision of the system considering only the first R results, so it considers the ranking of the results in its value.

$$R_p = \frac{r}{R} \tag{5.1}$$

where R is the number of relevant documents existing for the query and r is the number of relevant documents among the top-R retrieved documents.

2. **Mean Average Precision (MAP):** is the mean value across all queries of the average precision (AP) for each query. It averages the AP of every query in a complete evaluation.

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{5.2}$$

where Q is the number of queries and $AveP(q)$ is the average precision of query q. $AveP(q)$ adds the precision at every position of the ranking list containing a relevant document and averages the addition by the number of relevant results (for query q). It is defined as

$$AveP = \frac{\sum_{k=1}^{n}(P(k)rel(k))}{R} \tag{5.3}$$

where n is the number of retrieved documents, R is the number of relevant documents for query q, P(k) is the precision at cut-off k and $rel(k)$ is an indicator function taking value 1 if result at position k is relevant or 0 otherwise.

3. **Mean Reciprocal Rank (MRR):** is the multiplicative inverse of the rank of the first relevant result. It takes into account the first relevant result from the list and uses the inverse position to determine the performance of the system with regard to the query.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{5.4}$$

where $|Q|$ is the number of queries and $rank_i$ is the position of first relevant result of the i^{th} query.

4. **Normalized Discounted Cumulative Gain (NDCG):** measures the usefulness of a document based on its position in the result list. The gain is accumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks. This measure considers all relevant results, although the degree of relevance depends on the ranking of the results.

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{5.5}$$

where DCG at position p is defined in equation 5.6 and $IDCG_p$ is the ideal $nDCG$ at position p produced by ordering the results by decreasing relevance.

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2 i} \tag{5.6}$$

where rel_i is the relevance of i^{th} result and p is the position until which the cumulative gain is computed.

Table 5.5 shows the IR measurements of four configurations: three of them using each retrieval engine by itself and a fourth using the combination of every retrieval engine. Using a single retrieval engine means that the MIR system requests only one retrieval engine.

	QA	FTS	ObS	MultiEngine
MAP	0.085 (812,4%)	0.723 (-8,4%)	0.255 (133,2%)	0.720
MRR	0.092 (864,3%)	0.811 (10,7%)	0.261 (155%)	0.816
NDCG	0.101 (771,4%)	0.800 (7,2%)	0.268 (145,5%)	0.805

Table 5.5: IR measurements considering individual and multiple retrieval engines. The percentage gain between each RE and the multiengine approach developed in this thesis is shown in parenthesis.

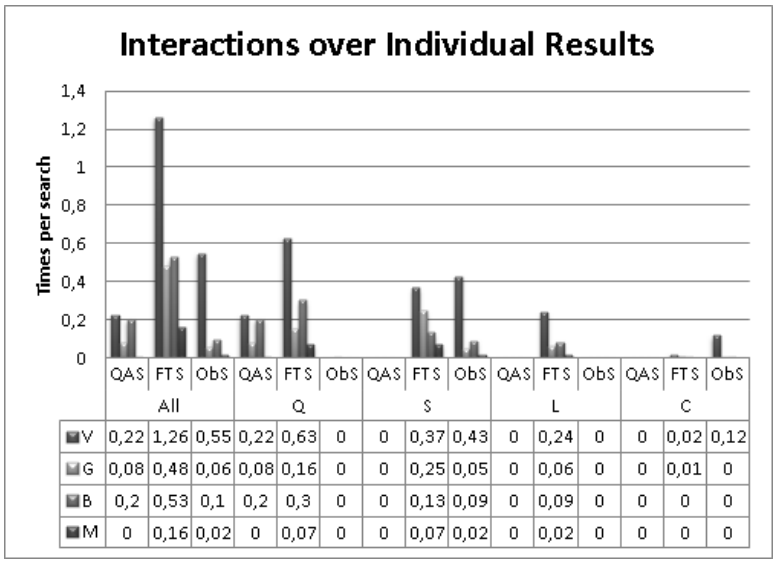
Evaluating each retrieval engine separately showed that results for *QAS* (`MAP = 0.085, MRR = 0.092 and NDCG = 0.101`) and *Ontology-based Search (ObS)* (`MAP = 0.255, MRR = 0.261 and NDCG = 0.268`) were poor. This happened because both RE were specific REs, i.e., *QAS* obtains accurate results for queries that are questions and *ObS* is more precise for exploratory search of concepts, so they were only useful for these types of query (questions and concepts respectively) and not for the rest of queries. The opposite happened with *FTS*, it was a system that supports every type of query and it was useful for all of them and got better results (`MAP = 0.723, MRR = 0.811 and NDCG = 0.8`).

Our *MultiEngine* system improved the results of every RE considered individually. MAP had a percentage loss of $8,4\%$, but on the contrary the other measurements got positive percentage gains: MRR had $10,7\%$ percentage gain and NDCG got $7,2\%$ increase.

5.3 Analysis of User Browsing

While users make use of the system, they visualize and judge different results for their queries. Exploratory analysis is divided into two parts: (1) the analysis of different types of documents viewed and judged, the considered documents are either those results where user has clicked on and those results that user has click on the faces accompanying it, and (2) the different visualizations that have been used (these visualizations can be: sorted list combining heterogeneous results, answers cloud, concepts grouping or list of specific results).

With regards to browsing and judgment of results, the measurements are averaged by search and showed classified by query text variant (question (Q), short text (S), long text (L), concept (C) and altogether (All)) and source (question answering (**QAS**), full-text (**FTS**) and ontology search (**ObS**)). The numerical results are shown in figure 5.1.



	QAS	FTS	ObS	QAS	FTS	ObS	QAS	FTS	ObS	QAS	FTS	ObS	QAS	FTS	ObS
	All			Q			S			L			C		
V	0,22	1,26	0,55	0,22	0,63	0	0	0,37	0,43	0	0,24	0	0	0,02	0,12
G	0,08	0,48	0,06	0,08	0,16	0	0	0,25	0,05	0	0,06	0	0	0,01	0
B	0,2	0,53	0,1	0,2	0,3	0	0	0,13	0,09	0	0,09	0	0	0	0
M	0	0,16	0,02	0	0,07	0	0	0,07	0,02	0	0,02	0	0	0	0

Figure 5.1: Mean number of browsed and judged documents per search. V symbolizes *'document visualizations'*, G refers to *'good relevance judgments'*, B is *'bad relevance judgments'* and M is *'neutral relevance judgments'*. Besides, x axis contains both query textual variants (question-Q, short-S, long-L and concept-C) and names of sources (question answering-QA, full text search-FTS and ontology-based search-ObS).

As Figure 5.1 details, the number of browsed or judged documents revealed four important issues to point out:

(1) Full-text search (*FTS*) results were more browsed and judged in almost all the query variants except for concept queries (*C*).

(2) No ontology (*Concept-ObS*) result was browsed or judged when a question (*Q*) was made and no answer (*QA* result) was browsed or judged when a short (*S*) or concept (*C*) query was sent.

(3) Only *FTS* results were browsed or judged when a long (*L*) query was made.

(4) Practically no judgments were made when a concept (*C*) query was used and almost all browsed or judged results were concepts.

The different visualization modes that are available in the GUI (see section 4.2.8) have been mapped to specific names to make results more readable. This names are:

- **List** refers to the visualization of results as a list of results. For example, in Figure 4.12 the results list is shown containing news documents and concepts of the ontology for the textual query *'Barcelona'*.

- **Doc** refers to the visualization of a single result.

- **Term List** refers to the visualization of concrete answers from QA retrieval engine as a list. Figure 4.13 depicts a list of answers for the question *'¿Quién es el presidente de la UEFA?' (Who is the president of UEFA?).*

- **Terms' Cloud** refers to the visualization of concrete answers from QA retrieval engine as a cloud of words. Figure 4.15 shows the cloud of answers for the question *'¿Quién es el presidente de la UEFA?' (Who is the president of UEFA?).*

- **Concepts' Cloud** refers to the visualization of ontology concepts as a cloud of words.

- **Concepts' Groups** refers to the visualization of ontology concepts semantically grouped. For example, in Figure 4.14 (down left) the list of semantic concepts is shown organized by their semantic category for the textual query *'Barcelona'*.

%	Answer List	Answer Cloud	Concepts Cloud	Concepts Groups	Doc
All	18,50	1,34	11,53	30,56	70,51
Q	28,95	4,39	0,00	28,07	71,93
S	15,00	0,00	16,88	31,88	69,38
L	13,33	0,00	0,00	13,33	80,00
C	9,62	0,00	30,70	46,15	63,46

Table 5.6: Percentage of queries that have led to use a concrete visualization mode. Query textual variants are represented by acronyms: *'question-Q'*, *'short-S'*, *'long-L'* and *'concept-C'*).

The percentage of queries that let to use a concrete visualization is shown in table 5.6. This table omits List visualization because every search began by showing a list of results.

Over 70% of queries (regardless of text variant) visualized **individual results**, while the use of special visualizations was not as widespread as expected. The **cloud of answers** (from QA) was not used at all (only 1.34% of queries). The **list of answers** was used in 18.50% of searches. On the contrary, the visualizations associated with concepts were more used: the **cloud of concepts** was used by 11.53% of queries while semantic **concepts grouping** reached 30.56%.

The figure also shows that when *short* (S) or *concept* (C) queries were used, the most frequently used views were concepts (cloud and grouping), while when querying with *questions* (Q) answer visualizations (cloud and list) were more used. This is consistent, but there were two cases where something different happened: with S and C queries, **Term (Answer) List visualization** was widely used (15% and $9,62\%$), and with Q queries the **concepts grouping** was high (28.07%). This is because these views were the first that were shown by 'other visualizations' menu button, i.e. if a short search was performed that returned concepts, and a user accessed the list of answers, (s)he did not observe that no answer exists until (s)he displayed the list.

5.4 Analysis of User Surveys

Although only $4,25\%$ of users filled out the survey, there were interesting results coming out from them. The survey questions are described in section A.1 (see annexe A). Questions were answered with a numbered value from 1 to 5, being 1 the minimum and 5 the maximum for each question. The averaged survey results for each question (with numeric result) are shown in Figure 5.2.

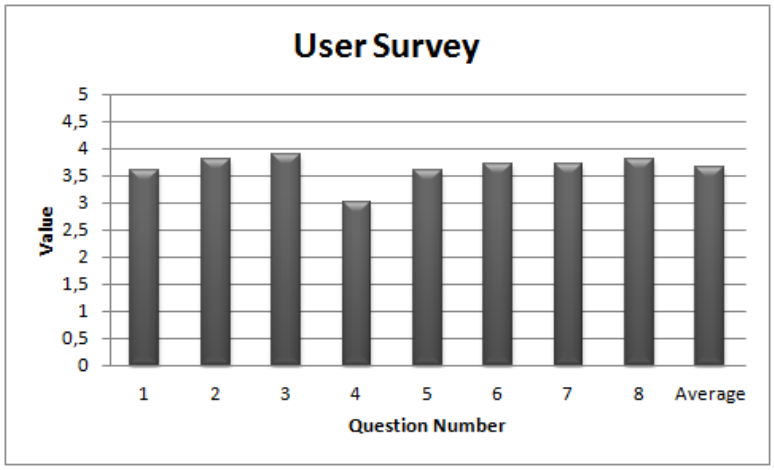


Figure 5.2: Results from the user survey analysis

As exposed in Figure 5.2, users evaluated the system with a mean value of $3,64$ over 5, being the lowest valued question the fourth one with a 3 and the most valued the third question with $3,9$ (see section A.1).

5.5 Discussion

The developed prototype was fully functional and it offered a wide range of search and visualization capabilities. Besides, the results of IR by applying expert-defined rules (in the handler module) improves the request of REs individually. The percentage gain obtained are: $771,4\%$ comparing *QAS*, $7,2\%$ comparing with *FTS* and $145,5\%$ comparing with *Concept-ObS*. The bad results of QAS and ObS are explained by its nature: they are specialized (query specific) REs, i.e. they are designed to work with a certain type of query, so they do not perform properly when queries of other types are

sent. Due to that, multiengine obtains huge percentage gains against them. On the contrary, since FTS is a nonspecific RE the gain is not so pronounced.

Multiengine obtains NDCG of $80,5\%$. This value is comparable with state-of-art systems such as those presented in the fedweb track (see section 2.8). Our system (multiengine) gets this result because we are combining systems that have high performance results.

6

Adapting IR functionality based on user interactions

Once the MIR model has been introduced (chapter 3) and the prototype implemented adopting this model has been described (chapter 4) and validated (chapter 5), the next step of this thesis is the definition of the techniques that will be used for adapting the functionality of the MIR system based on past user interactions. This fulfills the main goal of this thesis. Two types of information are considered for this adaptation: the query generated by the user and the past (previously performed) interactions (from every user).

Every component of the system can be modified in order to alter its functionality, but if we try to consider all the components at the same time, the number of variables makes it unaffordable. We believe that there are two components that can be reconfigured in order to obtain a better performance. These two elements are: (i) handler, modifying the rules that determine which retrieval engines are requested; and (ii) results' fusion module, changing the order in which the sources are combined.

The selection of elements (handler and results' fusion) is justified because there is a clear relation bewteen the performance of the retrieval and the two modules that manage which REs are requested and how results are combined. The IR functionality is modified by applying classification techniques, which will determine the requested REs and their order. The result we expect from this adaptation is the creation of new rules for the handler to lead to an improvement in information retrieval. This improvement will be measured through standard measures used in information retrieval: mean

average precision (MAP), mean reciprocal rank (MRR) and normalized discounted cumulative gain (NDCG).

6.1 Rule-Based Multimodal IR

The system has five components (query management, retrieval engines, retrieval engines handler, results management and interactions management) and it is unaffordable for the scope of this thesis to analyze and to modify the functionality of every component, so a selection has been done. The selection has focused on two components that are directly related to the retrieval engines: handler module and results' fusion module. This two modules are in charge of requesting every retrieval engine (based on the query) and combining the results retrieved from each retrieval engine. We have selected these modules because there are few works developing these modules in order to adapt them to the user behavior (previously performed interactions) as it has been reviewed in chapter 2. On the contrary, there are more works (as explained in section 2.7) studying user interactions or developing works on multimodal queries (see section 2.2).

The main idea of the functionality adaptation is depicted in figure 6.1. As can be seen, the past (previously performed) interactions are processed for generating a model, which modifies the functionality of handler and results' fusion module.

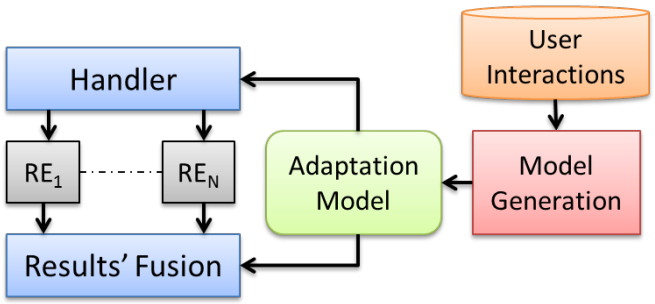


Figure 6.1: Schema of the functionality adaptation based on past interactions. Both handler and results' fusion modules are adapted using an interactions-based classification model.)

- The **Handler** requests multiple sources using a set of rules, which determine the ordered set of engines to be used for each query (as explained in section

3.2.4). These rules will be modified based on the past behavior of the user, i.e the previously performed interactions. As shown in figure 6.2, the handler's rules are created by a model generated using different classification algorithms. In the figure can be seen two handlers: the first handler (left side) without using the adaptation that generates a list of retrieval engines to request with the query; on the other hand (right side) the second handler generates a different list of retrieval engines.

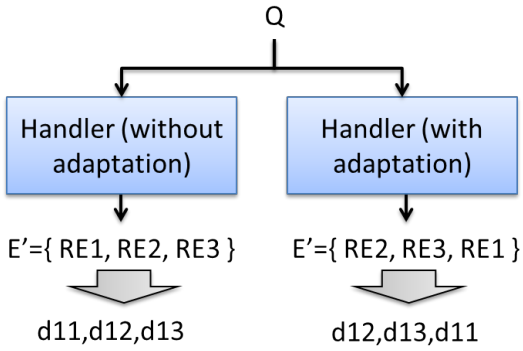


Figure 6.2: Results' fusion processing flow without and with functionality adaptation

The handler implemented in our prototype is based on rules. These rules are composed of two parts (see equation 6.1): conditions (left part), that must be met for the rule to be executed, and list of retrieval engines (right part), which are the engines that are requested with the query meeting the conditions.

$$conditions \rightarrow \mathcal{E}' = \{RE_1, \ldots, RE_Z\} \tag{6.1}$$

where $\mathcal{E}' = \{RE_1, \ldots, RE_Z\}$ represents an ordered list of retrieval engines that are requested if *'condition'* is met and $Z \leq N$ being N the number of available REs.

The rules use two types of information in the conditions: mode ($\mathcal{M}(Q)$) and type ($\Psi(Q)$) of the query, which values are shown in table 4.10.

$$\mathcal{M}(Q) = value \; and \; \Psi(Q) = value \rightarrow \mathcal{E}'\{RE_1, \ldots RE_h\} \tag{6.2}$$

where $\{RE_1, \ldots RE_h\}$ is the set of REs that are requested. If any retrieval source is not on the list it means that this source will not be requested when the rule conditions are met.

- Results' fusion implements a round robin algorithm. So the order in which the sources are requested directly influences the order in which the results are combined (as is shown in figure 6.2). So, depending on the list of retrieval engines, the order in which the results are combined is also influenced by the functionality adaptation.

The rules used by the handler of the basic prototype were manually defined using the query properties (see section 4.2.6). The modification of the functionality is based on the analysis of the users' past interactions which are analyzed and processed in order to generate new rules that represent user behavior. This analysis results in the generation of new rules that modify the functionality of the handler and fusion components. These rules are generated from past user interactions through classification models (section 6.1), query features (section 6.4.1) and retrieval engines' scores (section 6.4.2). The complete functionality of the adaptation is shown in figure 6.3.

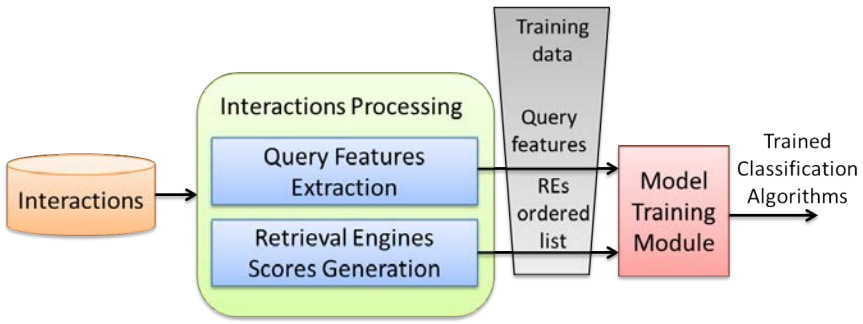


Figure 6.3: Schema of how the classification model for functionality adaptation is trained.

A cross-validation approach is used for evaluation. The interactions are divided randomly into 75% for training and 25% for testing.

6.2 Classification algorithms

As explained in Zukerman and Albrecht [2001], it is clear that there are plenty of machine learning approaches that can be used for analyzing users' behavior. From all

the possible algorithms we focus on three classification techniques in order to compare them. The selection of algorithms has been done due to the well-known efficiency of these algorithms for classification tasks. Therefore, we want to determine if they work properly for behavior pattern classification.

- Decision trees [Cintra et al., 2013] are build from a set of training data in the same way as ID3 [Quinlan, 1986], using the concept of information entropy. The training data is a set $S = s_1, s_2, ...$ of already classified samples. Each sample s_i consists of a p-dimensional vector $(x_{1,i}, x_{2,i}, ..., x_{p,i})$, where the x_j represent attributes or features of the sample, as well as the class in which s_i falls. In our case, the features of the vector are the characteristics of the query (mode, type, length, entities, etc.) while the class of each sample is an ordered list of retrieval engines. When requesting the model with a query classification information, the model will answer with the corresponding ordered list of *REs*.

Listing 6.1: Examples of input and output data for classification algorithms.

```
(qmode=t, qtype=long) -> ont, qa, ft
(qmode=t, qtype=question, qlength=14) -> qa, ft, ont
(qmode=t, qtype=short, qlength=2, qentities=alonso)->ont, qa, ft
```

- A multilayer perceptron [Gutiérrez et al., 2010] is an artificial neural network that maps input data to a set of output data. In our case, the multilayer perceptron is used for classification tasks where the input data is the information related to the query and the output data is the list of retrieval engines to be requested by this query. An MLP consists of multiple layers of nodes in a directed graph, with each fully connected to the next layer. Each node is a neuron (except the input nodes) with a nonlinear activation function. MLP uses a technique called back-propagation supervised learning to train the network.

- Simple K-Means [Kanungo et al., 2002] is a clustering method that divides a set of n observations into k groups in which each observation belongs to the closer group to the average. As shown in figure 6.4, the samples of the example are divided into three groups.

  In our case, each observation is composed of the query information and the requested *REs* list. This information is used to group and divide the observations.

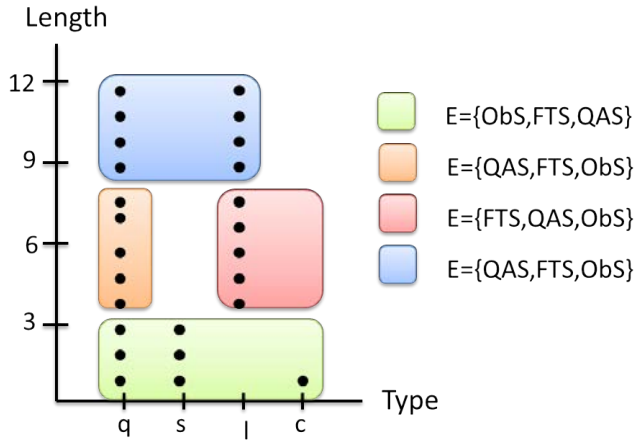


Figure 6.4: K-means classification algorithm example where four classes are generated

This can present a problem: if the number of groups is small, many queries are using the same *REs* list, while if there are too much groups, the algorithm will not generalize properly.

Decision trees, multilayer perceptron and K-means are used for classification tasks. The input of the algorithm is the user query and the past interactions. Meanwhile, the output of the algorithm is an ordered set of *REs*. The functionality of the algorithm is divided into three steps: analysis of input query, generation of interactions-based model and extraction of retrieval engines list.

6.3 Recording interactions with users

Although chapter 2 mentions some typical interactions that are recorded to analyze user behavior, this work is limited to a subset of the described interactions. Due to the lack of a standard that defines the interactions that must be registered, the approach presented in the work of Renaud and Azzopardi [2012] has been used and extended. The set of interactions taken from Renaud and Azzopardi [2012] are: password-controlled access, online questionnaires, study instructions and tutorials. Besides, other interactions have been also registered: performed queries, results browsing, relevance judgments, movements between visualizations and every action that is performed in the interface.

Moreover, the user's actions performed along the system are divided into sessions. A session begins when the user accesses the system and finishes when (s)he exits (pressing the log-out button) or closes the browser. The registered user actions can be classified in the following groups: (1) searches performed with the system, (2) buttons pressed and (3) displayed, browsed and judged results (relevant, neutral, or irrelevant).

Users must fill out a survey to find out their impressions about the system when they have finished using it (see section 4.2.8).

The interactions that are considered in this work are:

- **Registration and log:** every time that a user registers, logs in or logs out the system the interaction is registered.
- **GUI components pressed:** the components that are pressed by user navigation such as buttons, menus, etc. are registered together with the identifier of the component and the timestamp when the action took place.
- **Searches (queries and their associated information):** every time that a user performs a search, the query is stored directly and some information about it is analyzed and extracted. We propose in this work to store the following data:
  - *Mode of the query:* refers to the modality of the information composing the query between text (**t**), audio (**a**) and combination of text + image (**ti**).
  - *Type of the query:* determines the type of the corresponding textual part of the query (complete query for text mode, transcription for audio mode and textual part for combination mode). There are four possible values: short text, long text, question and concept.[112]
  - *Query content* depends on the format. It stores text or multimedia elements. Multimedia elements are stored in a dedicated server and are referenced by its URI.
- **Visualization of results:** the information related to results that users open (click on them in the list of results) is registered.

[112]Note the reader that "'concept"' is used in queries where the user searches for information related to a specific topic.

  - *Mode of the result:* there are three possible modes for the results: image (i), text (t) or video (v).
  - *Source* from where it has been obtained: the prototype offers three different sources: Question & Answering engine (QA), full text search engine (FTS) and ontology-based search (ObS).
  - *Position* of the result in the list of results.

- **Relevance feedback** of results: the prototype implements a relevance feedback mechanism to get feedback from users. This mechanism considers the following data:
  - *Mode of the result:* there are three possible modes for the results: image, text or video.
  - *Source* from where it has been obtained: the prototype offers three different sources: Question&Answering engine (QA), full text search engine (FT) and ontology search (ONT).
  - *Position* of the result inside the list of results.
  - *Relevance value* given to the result. Three possible values are considered:
    * Good: the result is relevant for the query.
    * Bad: the result is not relevant for the query.
    * Neutral: the result maybe relevant.

- **Survey**: a final survey to recruit the opinion that user had about the usability of the system was provided (see Annex A to see the questionnaire). The questionnaire has questions about usability of the system, types of queries that user has used, user satisfaction about visualization issues and personal opinion about the system.

The notation of the interactions is described in the formal model (see section 3.2.6). The different types of interactions ($\mathcal{T}$ in equation 3.21) considered during this evaluation and their associated information (Φ in equation 3.21) are:

1. $\mathcal{T} = REG$ representing the registration in the system.

2. $\mathcal{T} = LOG$ representing when a user logs in/out in the system. Its associated information contains the type of log action $\Phi \in \{IN,\ OUT\}$.

3. $\mathcal{T} = PRESS$ represents clicks, i.e. it registers each click that users do in the GUI. The identifier of the resource that has been clicked on $\Phi = \{\mathcal{W}\}$ is its associated information. $\mathcal{W}$ can be a button, an element of the list of results, a menu element or a relevance judgement element.

4. $\mathcal{T} = SEARCH$ represents the search actions. The query (text, audio file identifier or text and image file identifier) is the associated information: $\Phi = \{Q\}$.

5. $\mathcal{T} = VIEW$ represents the visualization actions. These actions are stored when a change of visualization mode happens. The new visualization mode identifier is the associated information, where there are four possible modes: $\mathcal{VI} \in$ {LIST, CLUSTER, GROUP, DOCUMENT}.

6. $\mathcal{T} = DOC$ represents an individual result visualization action. The associated information of this interaction is the identification of the viewed result: $\Phi = \{R_{i,j}\}$.

7. $\mathcal{T} = RELEV$ represents the relevance feedback actions, i.e. when a user judges a result. Its associated information is the identifier of the result ($R_{i,j}$) that has been judged and the value of the judgment ($\mathcal{VA} \in$ {GOOD, BAD, NEUTRAL}): $\Phi = \{R_{i,j}, \mathcal{VA}\}$.

An example of log file containing interactions of a complete search session is shown in figure 6.5. As can be seen, the session (with *sessionId=459*) started with *"'5151-login"'*, then user began a task (*"'5153-startTask"'*). Then (s)he performed a search (*5155-searchtxt*). At this point, some navigation of results' list (*"'5156, 5160"'*) and individual results (*"'5157, 5159, ..., 5163"'*) is done. Besides, all the pressing button actions are also registered (*5152, 5154, 5164, 5165*)

The logging process has been implemented inside the graphical user interface and the interactions are registered in a database.

The class diagram of the database is shown in figure 6.6. The database comprises 5 tables. The **Users** table stores user information (login and password), while **Personal** table stores the information provided during registration (date of birth, gender and

5151	459	0	login	2013-02-05 21:33:33	*NULL*
5152	459	0	pressbutton	2013-02-05 21:37:06	DoTask1
5153	459	0	starttask	2013-02-05 21:37:07	-tasknumber=1
5154	459	0	pressbutton	2013-02-05 21:37:14	textualsearch
5155	459	0	searchtxt	2014-04-21 15:50:52	¿Cuántos Kilómetros recorrió Samuel Sánchez en la ...
5156	459	0	viewlist	2013-02-05 21:37:20	pmf_05022013213714.xml-1-10
5157	459	0	viewdoc	2013-02-05 21:54:37	pmf_05022013213714.xml-1
5158	459	0	viewdoc	2013-02-05 21:55:06	pmf_05022013213714.xml-2
5159	459	0	viewdoc	2013-02-05 21:56:35	pmf_05022013213714.xml-2
5160	459	0	viewlist	2013-02-05 21:56:45	pmf_05022013213714.xml-1-10
5161	459	0	viewdoc	2013-02-05 21:57:02	pmf_05022013213714.xml-3
5162	459	0	viewdoc	2013-02-05 21:57:06	pmf_05022013213714.xml-4
5163	459	0	viewdoc	2013-02-05 21:57:11	pmf_05022013213714.xml-5
5164	459	0	pressbutton	2013-02-05 21:57:32	goback
5165	459	0	pressbutton	2013-02-05 21:57:53	goexit
5166	459	0	logout	2013-02-05 21:57:53	*NULL*

Figure 6.5: Interactions of a complete session example

work position). The **Sessions** table stores information about sessions made by users on the system. **Interactions** table stores the interactions that users perform storing interaction identifier, session identifier and interaction type but also logging the timestamp and an additional information field. The last table (**PostSessForm**) stores the information provided by users when the user survey is filled out.

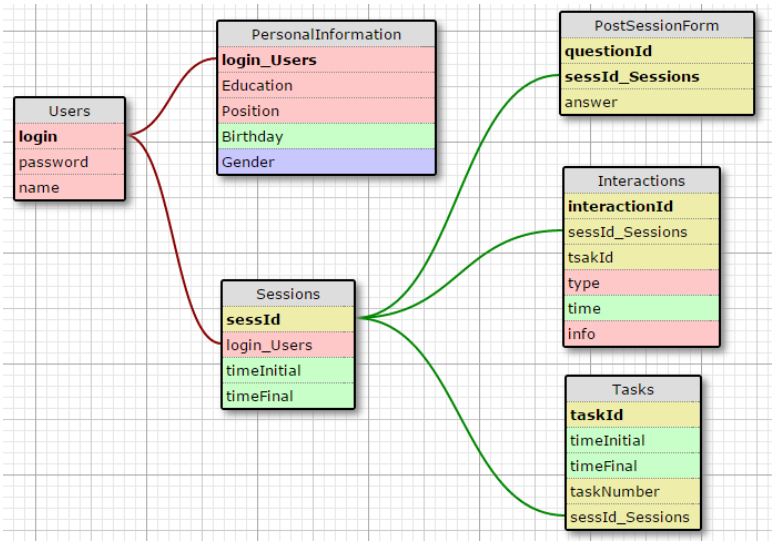


Figure 6.6: Class Diagram of the database that stores user interactions

6.4 Preparing training data for models generation

Classification models (decision trees, MLP and K-means) will be used to classify queries and obtain rules for the handler. Before we can use the models, these must be generated (trained), for which we need a set of previously classified data.

Once we have the interactions, we have to analyze and process them to turn in the proper format required for models. Rules-generation models need a training set to classify future queries. A classified data is composed of two parts: a set of features that describe this data and the classification that should be assigned to it. In our case, a classified data is composed of the query features and the order in which the retrieval engines is requested. Once the model is created, it will be used to classify new queries. For this we need two things: information related to the query (see section 6.4.1) and the list of REs must contain the rule associated with that query (see section 6.4.2).

6.4.1 Query Analysis and Statistics

The functionality adaptation of the MIR system is based on classification algorithms which will decide the engines to be requested with each query based on the past (previously performed) interactions and information of the query.

The classification algorithms need some training data, which are composed of the information related to the query and the order of REs to request.

We will focus on their linguistic characteristics.

We have used the following characteristics as a first step to determine which one of them is more relevant for the IR functionality adaptation.

1. **Mode (m)**: the mode of the query between *'t' (text)*, *'a' (audio)* or *'ti' (text and image)*.

2. **Type (t)**: the type of the query between *'Question'*, *'Short'*, *'Long'* or *'Concept'* (see section 5.1).

3. **Length (l)**: number of tokens of the query.

4. **Named Entities in the query (e)**. In this approach we decided to use information about named entities in queries. These named entities will be PERSON,

LOCATION, ORGANIZATION, etc. These entities are extracted with the commercial tool MeaningCloud[113].

5. **Number of named entities (n_e)** present in the query analyzed with the MeaningCloud tool.

6. **Number of verbs (n_v)** analyzed using the MeaningCloud Part-Of-Speech tagger[114].

7. **Topic (o)**: topic of the query extracted using MeaningCloud Topics Extraction[115].

Next (in figure 6.7) is shown a graphical query example together with its characteristics.

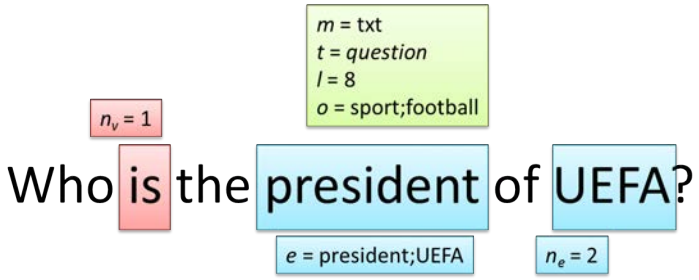


Figure 6.7: An example showing a Question as query with the features described in section 6.4.1

6.4.2 Defining the ranking (scores) of the retrieval engines

The second part of the input for the classification algorithms is the list of retrieval engines requested with every query. The prototype had predefined rules, so the requested REs were the same for the same type of queries. Because of that, we need to find these REs that have been more useful for users (based on interactions).

The recorded interactions during the prototype evaluation are the input for the AI algorithm. So, in order to generate a model using WEKA, data input needs to have a concrete format.

[113]http://www.meaningcloud.com/ accessed at 23/07/2015
[114]https://www.meaningcloud.com/developer/lemmatization-pos-parsing accessed at 23/07/2015
[115]http://www.meaningcloud.com/products/topics-extraction/ accessed at 23/07/2015

The problem is that the classification information (order of *REs* for each query) must be provided by an expert. We do not have resources enough to create this expert classification, so the list of ordered retrieval engines for each query is computed using a set of scores.

We base our approach on the works of Balog [2013]; Pal and Mitra [2013] that define the score of a retrieval engine as a linear combination of the scores of three aspects: (1) context, referring to the environmental characteristics of the retrieval engine; (2) content, referring to the similarity between the content of the collections used by the retrieval engine; and (3) past users behavior, refering to the actions (recorded as interactions) which have been previously performed by users while using the system. This linear combination is shown in equation 6.3.

$$\begin{aligned} \alpha_i = score(RE_i, Q) = \quad & w^i_{context} \cdot score^i(context, Q) + \\ & w^i_{content} \cdot score^i(content, Q) + \\ & w^i_{behavior} \cdot score^i(behavior, Q) \end{aligned} \tag{6.3}$$

where Q is the query sent by the user, $1 \leq i \leq N$, N being the number of retrieval engines, w^i_x is the weight of each factor for i^{th} retrieval engine, $\sum_x w^i_x = 1$ and $score^i(x, Q)$ is the score of RE_i for each factor.

Our approach considers only the past behavior, so the simplified equation is shown in equation 6.4.

$$\alpha_i = score(RE_i, Q) = w^i_{behavior} \cdot score^i(behavior, Q) \tag{6.4}$$

where $w^i_{behavior} = 1$.

In our case, the user behavior is defined by the previously performed interactions, so the value obtained for each source based on these interactions is defined in equation 6.5.

$$score^i(interactions, Q) = score^i_Q = \frac{\sum_{j=1}^{N} score(d_{ij}, Q)}{N} \tag{6.5}$$

where $score(d_{ij}, Q)$ is the score of a document d_{ij} with respect to the input query Q and N is the number of considered documents.

Different scores of documents are considered to generate the retrieval engines order for the training data which will be later used for the classification algorithms:

- Score based on the total number of logged interactions of each document.

$$score(d_{ij}, Q) = \frac{G}{K} \tag{6.6}$$

where G is the number of "good" relevance judgments of document d_{ij} and K is the total number of interactions done during the search with query Q.

- **Interactions-based score** only considers a document as relevant if an interactions over it has taken place.

$$score(d_{ij}, Q) = \begin{cases} 1 \; if \; interaction \; exists \; over \; doc \; d_{ij} \\ 0 \; otherwise \end{cases} \tag{6.7}$$

- RE-Score-based score: the score of each document is directly the score given by the RE.

$$score(d_{ij}, Q) = score_{RE_i}(d_{ij}, Q) \tag{6.8}$$

where $score_{RE_i}$ is the score of document d_{ij} directly obtained from RE_i.

- **Lowest-position score** is based on the position of reviewed results with the lowest position within the results' list. The value of each retrieval engine is computed using equation 7.2.

$$score^i_Q = \min_j \frac{1}{rank(d_{ij}, Q)} \tag{6.9}$$

where $rank(d_{ij}, Q)$ is the ranking of d_{ij} within the results' list.

- Ranking-based or **Average-position-based score**: it uses the ranking of the document as a relevance indicator.

$$score(d_{ij}, Q) = \frac{1}{rank(d_{ij}, Q)} \tag{6.10}$$

where $rank(d_{ij}, Q)$ is the ranking of document d_{ij} in the results' list from RE_i.

- Iteration-based score.

$$score(d_{ij}, Q) = \frac{1}{iteration(d_{ij}, Q)} \tag{6.11}$$

where $iteration(d_{ij}, Q)$ is the iteration (number of user actions) in which the document d_{ij} has been used. It takes value $iteration(d_{ij}, Q) = 1$ if it is the first result viewed or marked, $iteration(d_{ij}, Q) = 2$ if it is the second and so on.

- The **mathematical score** considers both the position of reviewed results and the iteration in which they have been reviewed.

$$score(d_{ij}, Q) = \frac{1}{1 + rank(d_{ij}, Q)} \cdot \frac{1}{\log(1 + iteration(d_{ij}, Q))} \tag{6.12}$$

  where $rank(d_{ij}, Q)$ is the ranking of d_{ij} within the results' list and $iteration(d_{ij}, Q)$ is the number of interactions made over d_{ij}. This equation has been taken from Womser-Hacker [1996] and we adapted it adding $log(\cdot)$ to consider also the decreasing of not been the first 'used' result.

Once the scores for every RE have been computed, they are put together to generate an ordered list, which is the classification information (together with the query information) used as input (training data) for the classification algorithms.

Since we have described in detail the algorithm that we apply to improve the results of IR, now we have to evaluate this algorithm to quantify the improvement that can be obtained with different configurations presented.

7

Experimental setups of IR adaptation based on user interactions

This chapter is devoted to describe the experiments carried out to validate the adaptation of IR functionality based on user interactions developed in chapter 6. In order to validate and compare the different algorithms and techniques described in chapter 6, we propose to compare the performance, normalized discounted cumulative gain (NDCG) measure, of an IR system using different configurations (decision trees, K-means, different scores for determining the REs orders, etc.) by applying the same queries.

7.1 Experiment design for IR adaptation algorithm evaluation

This evaluation is intended to validate the different variations of the functionality adaptation applied to the interactions collected during the evaluation of the prototype. As indicated in chapter 5 the prototype with the baseline approach (or rules defined by experts) was online for eight weeks compiling interactions. These interactions (as explained in section 6.3) reflect user behavior when searching and reviewing the results.

These interactions have been used to generate models (decision trees, multilayer perceptron and K-means) that are used for generating rules for the handler (see section 6.1).

The validation of the functionality adaptation is defined as a Cranfield experiment [Project and Cleverdon, 1962]. Figure 7.1 displays the methodology we use for the experiment describing each of its four steps:

1. First of all a nomenclature is defined in order to simplify the reading of the obtained results (see section 7.2).

2. A cranfield experiment is characterized by using a goldstandard (as explained in section 5.2), but in our case there is no goldstandard available, so the second step is the definition of a silver standard corpus (see section 5.2).

3. The techniques (scores for ranking retrieval engines) and algorithms (classification models) described in chapter 6 are applied to the interactions defined in the silver standard. By applying these techniques and algorithms several set of rules for the handler are obtained (see section 7.3).

4. The analysis of the results is the last step of the experiment. This analysis is done by comparing the normalized discounted cumulative gain (NDCG) value of each approximation (see section 7.4).

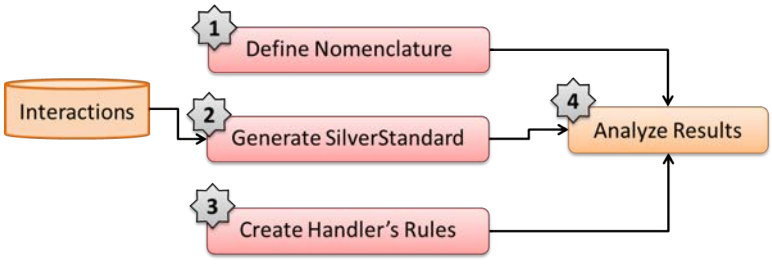


Figure 7.1: Methodology of the experiment for validating the functionality adaptation.

7.2 Clarifying nomenclature

As explained in chapter 6, the defined functionality adaptation generates a model using decision trees, multilayer perceptron and k-means which are fed using two types of information: the features of the queries and the associated ordered list of requested *REs*.

The query's features have been described in section 6.4.1 and decision tree, multilayer perceptron and k-means were already presented in section 6.1. The retrieval engines' scores are defined in section 6.4.2. Therefore, in order to simplify the readability of the results, acronyms are assigned to each classification algorithm, REs score and query feature.

- Rules generation classification algorithms (see section 6.2):
  - ***Prototype*** determines the basic prototype using predefined rules in the handler.
  - ***Probs*** is a simple probability-based method. It does not use any classification technique or algorithm. It considers the probabilities of use of every source, i.e., every source receives a score equal to the total number of interactions related to its results divided by the total number of interactions.
  - The C4.5 decision tree algorithm is labeled as ***J4.8***.
  - The multilayer perceptron technique is refered as ***MLP***.
  - The simple K-means (2 groups) algorithms is named as ***SKM2***.
- Query features (see section 6.4.1):
  - **m**: *mode* of the query.
  - **mt**: mode and *type* of the query.
  - **mtl**: mode, type and *length* of the query in number of tokens.
  - **mtle**: mode, type, length and *textual entities*.
  - **mtleNe**: mode, type, length, textual entities and *number of entities*.
  - **mtleNeNv**: mode, type, length, textual entities, number of entities and *number of verbs* contained in the query.

  - **mtleNeNvT**: mode, type, length, textual entities, number of entities, number of verbs and *topic* of the query.

- Retrieval engines ranking scores (see section 6.4.2):
  - **IbS**: interaction-based score.
  - **LPS**: lowest-position score.
  - **APS**: average-position-based score.
  - **FUS**: first-used score.
  - **Maths**: mathematical score.

7.3 IR adaptation based on user interactions

The classification algorithms presented in section 6.2 are applied through the WEKA[116] tool.

The rules obtained by querying the model have the format defined in equation 3.12. These rules have been added to a text file that is currently online at: `http://sphynx.uc3m.es/resources/HandlerRules.xml`. This file contains a set of the algorithm configuration (classification techniques, scores and query information) together with the generated rules for each configuration. Each rule contains the query information it is triggered with and the list of retrieval engines that should be requested with this query (its result).

The rules of the model are obtained by requesting the model with every possible query feature and obtaining the corresponding *REs* ordered list. This means that in case the query has the features considered in the left part of the rule, then the RE appearing in the right part of the rule will be executed. Since the number of possible combinations of features and REs rankings is very large, we can not explain every set of rules. Next are described some representative combinations. The rules are ordered by the features of the query, so they do not have any relevance value.

The first rules displayed (see table 7.1) are those obtained by the model generated using decision trees (**'J4.8'**) as classification algorithm, the mode of the query (**'m'**) and the *REs* ranking determined by the first-used score (**'FUS'**). As can be seen, there

[116] http://www.cs.waikato.ac.nz/ml/weka/ accessed at 23/07/2015

are only two rules, one for each query features possibility. Each rule returns a different *REs* order. These rules represent that when a text query is presented the sequence of RE is first full text then question answering and finally ontology service.

```
qmode=t; -> ft,qa,ont
qmode=ti; -> qa,ft,ont
```

Table 7.1: Rules obtained by decision trees (**'J4.8'**) with the mode of the query (**'m'**) and the *REs* ranking determined by the first-used score (**'FUS'**).

Analyzing these rules (table 7.1) it can be seen that when the query is text only, we first request the FTS engine, then QAS and finally the ObS. This is consistent with the fact that most of the text queries reviewed results are textual documents and not concepts nor specific answers. When it comes to a combined query (text and image), the text is usually a question that seeks to complement the information of the image. Therefore, the most consulted results are concrete answers and it is therefore logical that the model has placed this engine first.

The second displayed set of rules (see table 7.2) are those obtained by the model generated using decision trees (**'J4.8'**) as classification algorithm, the mode and type of the query (**'mt'**) and the *REs* ranking determined by the first-used score (**'FUS'**). As can be seen, there are some rules that return the same ordered list.

```
qmode=t;qtype=question; -> qa,ft,ont
qmode=t;qtype=short; -> ont,qa,ft
qmode=t;qtype=long; -> ont,qa,ft
qmode=t;qtype=concept; -> ont,qa,ft
qmode=ti;qtype=long; -> ft,qa,ont
qmode=ti;qtype=question; -> ft,qa,ont
qmode=ti;qtype=short; -> ont,qa,ft
```

Table 7.2: Rules obtained by decision trees (**'J4.8'**) with the mode and type of the query (**'mt'**) and the *REs* ranking determined by the first-used score (**'FUS'**).

When queries are questions the first RE it requests is **QAS** because a greater number of exact answers will be displayed or ranked as relevant, while the latter is **ObS** because no results thereof are offered. The opposite happens with short queries,

ObS is the first and **QAS** is the last. The strange case is *long queries*, which should request first **FTS** engine, but it does last.

When users give a combined query (question with a picture) the first engine is **FTS** instead of **QAS**. This is because of the functionality of the handler with combined queries. The text part of the query is joint to the text extracted from the image (objects and over impressed text) before sending it to the textual REs. So, when users send a question, it is joined to the extracted image text and its structure changes from question to long text.

Finally, it is interesting that the case of combined query (*ti*) where the text has *concept* type is not considered. The model does not return a rule for this case because this case is not present in the training data. Besides, it makes no sense to request the system with a concept together with and image, so concepts are used for exploratory search only.

The third set of rules displayed (see table 7.3) are those obtained by the model generated using simple K-means (**'SKM2'**) as classification algorithm, the mode, type and length of the query (**'mtl'**) and the *REs* ranking determined by the first-used score (**'FUS'**).

```
qmode=t;qtype=question;qlength=8; -> qa,ft,ont
qmode=t;qtype=question;qlength=14; -> qa,ft,ont
qmode=t;qtype=question;qlength=6; -> qa,ft,ont
qmode=t;qtype=short;qlength=1; -> ont,qa,ft
qmode=t;qtype=short;qlength=2; -> ont,qa,ft
qmode=t;qtype=short;qlength=3; -> ont,qa,ft
qmode=t;qtype=long;qlength=5; -> ont,qa,ft
qmode=t;qtype=long;qlength=6; -> qa,ft,ont
qmode=t;qtype=concept;qlength=1; -> ont,qa,ft
qmode=ti;qtype=long;qlength=4; -> ft,qa,ont
qmode=ti;qtype=long;qlength=7; -> ft,qa,ont
qmode=ti;qtype=question;qlength=8; -> ft,qa,ont
qmode=ti;qtype=short;qlength=2; -> ont,qa,ft
qmode=ti;qtype=short;qlength=1; -> ont,qa,ft
```

Table 7.3: Rules obtained using simple K-means (**'SKM2'**) with the mode type and length of the query (**'mtl'**) and the *REs* ranking determined by the first-used score (**'FUS'**).

The same conclusions as in the previous case can be drawn from these rules (table 7.3) except for one case: when the query is only text and long (more than three words). In this case there is a difference depending on the length of the query.

The last rules displayed (see table 7.4) are those obtained by the model generated using multilayer perceptron (**'MLP'**) as classification algorithm, the mode, type, length and entities of the query (**'mtle'**) and the *REs* ranking determined by the mathematical score (**'Maths'**).

```
qmode=t;qtype=long;qlength=4;qentities=Mundial_de_F1;->ont,qa,ft
qmode=t;qtype=long;qlength=5;qentities=none;->ont,qa,ft
qmode=t;qtype=question;qlength=6;qentities=Miguel_Garcia;->qa,ft,ont
qmode=t;qtype=short;qlength=2;qentities=fernando_alonso;->ont,qa,ft
qmode=t;qtype=concept;qlength=1;qentities=none;->ont,qa,ft
qmode=t;qtype=question;qlength=7;qentities=Pau_Gasol;->qa,ft,ont
...
qmode=ti;qtype=question;qlength=8;qentities=none;->ft,qa,ont
qmode=ti;qtype=long;qlength=7;qentities=none;->ft,qa,ont
qmode=ti;qtype=short;qlength=2;qentities=David_Washington->ont,qa,ft
```

Table 7.4: Rules obtained using multilayer perceptron (**'MLP'**) with the mode type length and entities of the query (**'mtle'**) and the *REs* ranking determined by the mathematical score (**'Maths'**).

A clear conclusion can be drawn from the rules in table 7.4: the entities have no relevance to the order of REs. This is clearly seen in one case: there are two queries that have text mode and long type, but the entities are different among them (one has an entity while the other has none). These two cases return the same order of retrieval engines, so the entities have no influence in the order.

A set of rules has been generated for each combination of algorithms and techniques presented in chapter 6. None of the generated rules' sets has been directly integrated into the prototype. The main reason for not integrating the rules is that the integration of the rules made mandatory to perform another evaluation using final users. Instead of integrating them, we have evaluated the generated rules without using final users.

7.4 Analysis of results when applying different approaches

Once the rules have been obtained for every possible combination of the adaptation algorithm, we proceeded to assess them. Due to the fact that the new rules have not been integrated in the prototype, the evaluation process was carried out following the next three steps:

1. First of all, the corpus to evaluate the rules is selected. In this case, the silverstandard corpus defined in section 5.2 is used.

2. A set of the silverstandard corpus has been selected, i.e., a crossvalidation approach is adopted. 75% of the search sessions (interactions associated to them) are used for training the models (and generating the rules) and 25% of the search sessions are used for evaluation. The selection is done by dividing randomly the available search sessions of the silverstandard.

3. Once the evaluation data has been extracted, the performance of the rules is measured by normalized discounted cumulative gain (NDCG) measure. We use this measure because it considers not only the relevance of a result but also its ranking in the results' list.

The results are organized according to the *REs* ranking score in the next sections.

7.4.1 Interactions-based score (IbS)

The interaction-based score approach is based on the interactions logged from the users. The value of each retrieval engine is computed using equation 6.5 where N is the number of retrieved documents and

$$score(d_{ij}, Q) = \begin{cases} 1 \; if \; interaction \; exists \; over \; doc \; d_{ij} \\ 0 \; otherwise \end{cases} \tag{7.1}$$

The numeric results (*NDCG*) are shown in table 7.5 and in figure 7.2 for every classification algorithm and query property.

The results for these combinations of the algorithm offer little improvements over the baseline. The best result is obtained using the multilayer perceptron classification algorithm (**MLP**) and the mode, type, length, entities, number of entities and number

Algorithm	m	mt	mtl	mtle	mtleNe	mtleNeNv	mtleNeNvT
Prototype	79.31						
Maths	79.22	79.38	80.72	80.53	79.62	80.61	80.71
J48	79.21	80.59	80.71	80.24	80.72	80.44	80.33
MLP	80.34	80.64	80.3	80.13	80.84	81.38	80.78
SKM2	76.99	79.3	80.05	80.77	79.68	79.95	80.17

Table 7.5: NDCG for machine learning algorithm and query types using **interaction-based** rules-generation approximation

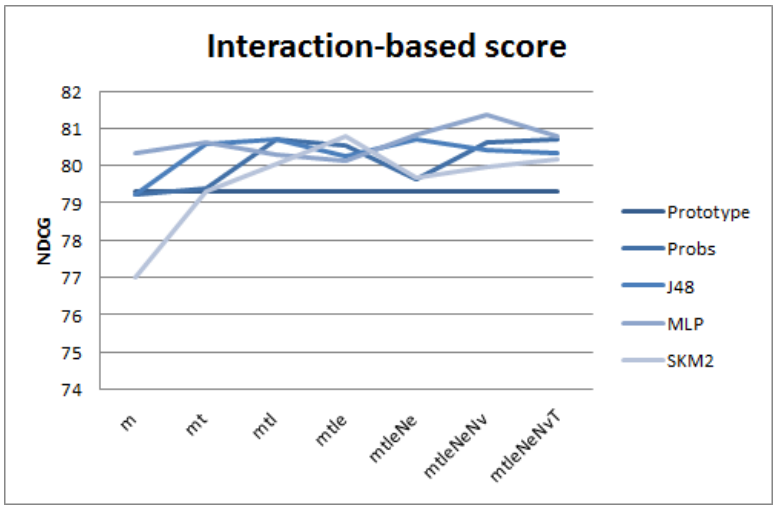


Figure 7.2: NDCG measurements for machine learning algorithm used and query types using **interaction-based score** rules-generation approximation

of verbs of the query (**mtleNeNv**). Its NDCG value is 81,38%. It seems to be a high value but the percentage gain against the baseline is only 2,61%. The small percentage gain is due to the fact that the retrieval engines offer relevant results in the top positions of their results list, so the modification of the order they are requested has no big influence in the final NDCG value.

It is also interesting to note that there are some combinations worse than the baseline. These cases occur when the query information is too simple: only the mode or the mode and type. Queries' classification fails because the information on the query

is too generic and the model classifies as similar very different queries: two completely different queries are a question and a concept, but both have the same mode (text).

7.4.2 Lowest-Position score (LPS)

The lowest-position score approach is based on the position of reviewed results with the lowest position within the results' list. The value of each retrieval engine is computed using equation 7.2.

$$score_Q^i = \min_j \frac{1}{rank(d_{ij}, Q)} \tag{7.2}$$

where $rank(d_{ij}, Q)$ is the ranking of d_{ij} within the results' list.

The numeric results ($NDCG$) are shown in table 7.6 and in figure 7.3.

Algorithm	m	mt	mtl	mtle	mtleNe	mtleNeNv	mtleNeNvT
Prototype	79.31						
Probs	78.38	80.96	80.08	80.13	80.46	80.84	79.95
J48	79.96	80.02	80.21	80.07	80.52	81.21	80.35
MLP	79.66	79.52	80.06	81.05	80.63	80.91	80.15
SKM2	77.87	80.59	80.58	79.46	80.5	79.55	80.2

Table 7.6: NDCG for machine learning algorithm and query types using **lowest-position** rules-generation approximation

These results also offer little improvements over the baseline. The best result is obtained using the decision tree classification algorithm (**J48**) and the mode, type, length, entities, number of entities and number of verbs of the query (**mtleNeNv**). Its NDCG value is $81,21\%$. Although it is a high NDCG value, the percentage gain against the baseline is only $2,4\%$. Just to mention it, the second best result is obtained using the multilayer perceptron classification algorithm (**MLP**) and the mode, type, length and entities of the query (**mtle**) and it obtains a percentage gain (against the baseline) of $2,2\%$. The small difference between both percentage gains claims that there is no specific combination that works much better than the others.

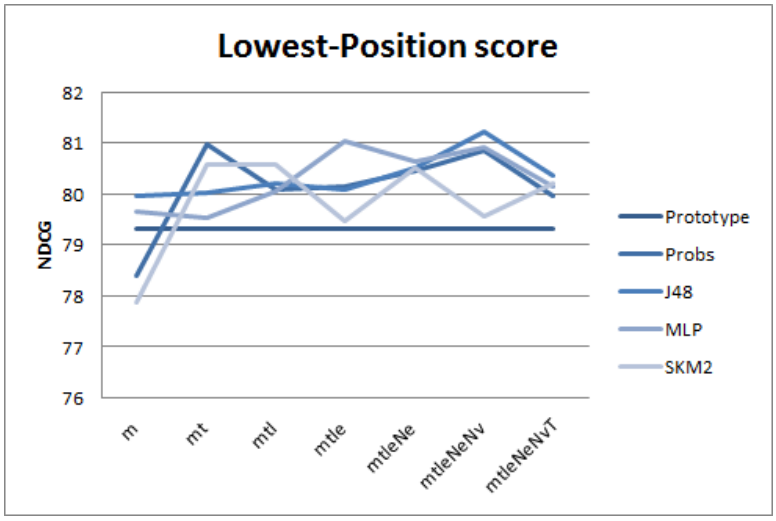


Figure 7.3: Graphical display of NDCG for machine learning algorithm and query types using **lowest-position** rules-generation approximation

It can only be remarked that when little information is used in the query: mode, type and length, the results are worse than when the query information is more specific. This indicates that as best the model can sort the query, the better the results and thus better are the rules. This happens through last case. When the topic is added to the results become worse. This may be due to two reasons: the topics are not well allocated and are introducing noise or the topics are so generic that spoil the classification of the queries.

In this case there are only two combinations worsening baseline results. The reason for this deterioration is the same as in the previous case.

7.4.3 Averaged-Position score (APS)

The averaged-position score approach is based on the position of every reviewed result within the results' list. The score of each retrieval engine is computed using equation 6.5 where N is the number of considered (viewed and marked) results and

$$score(d_{ij}, Q) = \frac{1}{rank(d_{ij}, Q)} \tag{7.3}$$

where $rank(d_{ij}, Q)$ is the ranking of d_{ij} within the results' list.

The numeric results (limited only to NDCG) are shown in table 7.7 and in figure 7.4.

Algorithm	m	mt	mtl	mtle	mtleNe	mtleNeNv	mtleNeNvT
Prototype	79.31						
Probs	80.31	80.84	80.03	81.54	79.73	80.83	80.05
J48	79.06	80.02	80.16	80.67	80.7	80.38	0.0
MLP	78.83	80.56	80.65	79.89	80.38	80.76	80.06
SKM2	78.65	80.71	80.58	79.77	79.22	79.54	79.6

Table 7.7: NDCG for machine learning algorithm and query types using **averaged-position** rules-generation approximation

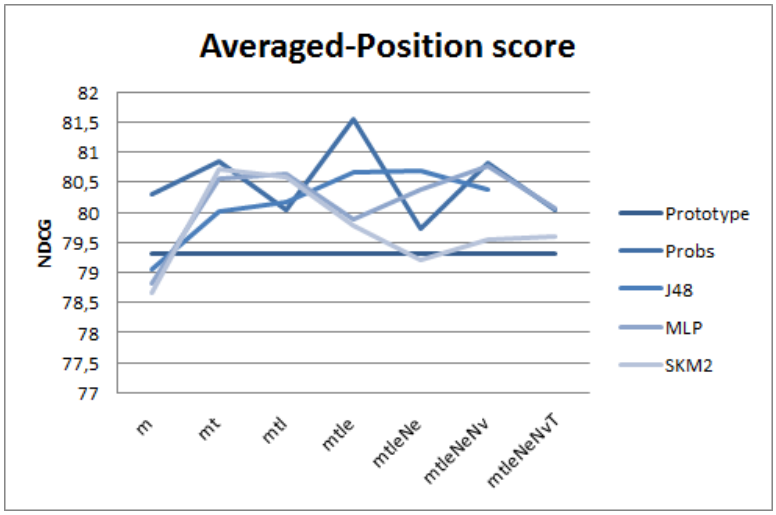


Figure 7.4: NDCG measurements for machine learning algorithm and query types using **averaged-position** rules-generation approximation

As in the previous case, the results offer little improvements over the baseline. The best result is obtained using probabilities-based classification algorithm (**Probs**) and the mode, type, length and entities of the query (**mtle**). Its NDCG value is $81,54\%$. Although it is a high NDCG value, the percentage gain against the baseline is only $2,81\%$. The second best result has a value a little lower than the first, but the difference

between the NDCG value of the best and the NDCG value of the second is bigger as in the previous combinations of the algorithm.

In this case there are also four combinations worsening baseline results. The reason for this deterioration is the same as in the previous cases. The last thing that can be noted is that there is a case that did not return results, possibly due to a problem of execution during the evaluation. This issue should be better studied if finally opt to add this classification algorithm and the query information.

7.4.4 First-Used score (FUS)

The first-used (*FUS*) score assigns a score to each engine based on the order in which the results from this engine has been requested, i.e if the first revised result comes from eng_1 and the second comes from eng_3, the order of engines is eng_1, eng_3.

The numeric results (limited only to NDCG) are shown in table 7.8 and in figure 7.5.

Algorithm	m	mt	mtl	mtle	mtleNe	mtleNeNv	mtleNeNvT
Prototype	79.31						
Probs	79.59	79.93	80.71	80.75	80.25	79.82	80.43
J48	79.23	80.36	79.42	80.71	80.86	80.26	80.8
MLP	79.58	79.58	80.77	80.69	79.76	80.57	79.74
SKM2	78.3	80.65	80.52	79.07	79.68	80.07	79.44

Table 7.8: NDCG for machine learning algorithm and query types using **first-used** rules-generation approximation

The results for these combinations of the algorithm also offer little improvements over the baseline. The best result is obtained using the decision tree classification algorithm (**J48**) and the mode, type, length, entities and number of entities of the query (**mtleNe**). Its NDCG value is $80,86\%$.

In this case there are only two combinations worsening baseline results. The reason for this deterioration is the same as in the previous cases.

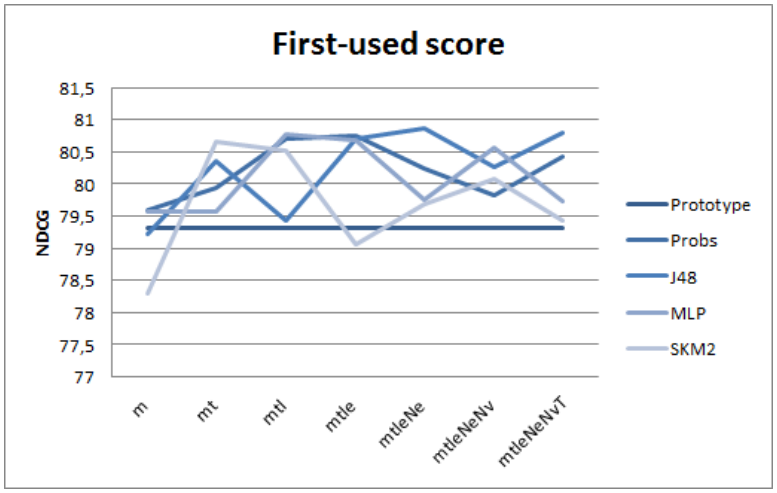


Figure 7.5: NDCG measurements for machine learning algorithm and query types using **first-used** rules-generation approximation

7.4.5 Mathematical score (Maths)

The mathematical score considers both the position of reviewed results and the iteration in which they have been reviewed. The value of each retrieval engine in this approach is computed using equation 6.5 where N is the number of considered (viewed and marked) results and

$$score(d_{ij}, Q) = \frac{1}{1 + rank(d_{ij}, Q)} \cdot \frac{1}{\log(1 + iteration(d_{ij}, Q))} \tag{7.4}$$

where $rank(d_{ij}, Q)$ is the ranking of d_{ij} within the results' list and $iteration(d_{ij}, Q)$ is the number of interactions made over d_{ij}. This equation has been taken from Womser-Hacker [1996] and we adapted it adding $log(\cdot)$ to consider also the decreasing of not been the first 'used' result.

The numeric results (limited only to NDCG) are shown in table 7.9 and in figure 7.6.

These combinations of the algorithm also offer little improvements over the baseline. The best result is obtained using decision tree classification algorithm (**J48**) and the whole query information (mode, type, length, entities, number of entities, number of verbs and topic) (**mtleNeNvT**). Its NDCG value is $81,33\%$.

Algorithm	m	mt	mtl	mtle	mtleNe	mtleNeNv	mtleNeNvT
Prototype	79.31						
Probs	79.85	80.28	81.18	80.29	80.05	80.15	79.79
J48	79.12	80.02	80.1	79.59	79.68	80.52	81.33
MLP	79.23	80.43	80.37	80.78	80.28	80.38	80.9
SKM2	78.28	78.47	80.13	79.39	79.58	80.18	80.49

Table 7.9: NDCG for machine learning algorithm and query types using **mathematic** rules-generation approximation

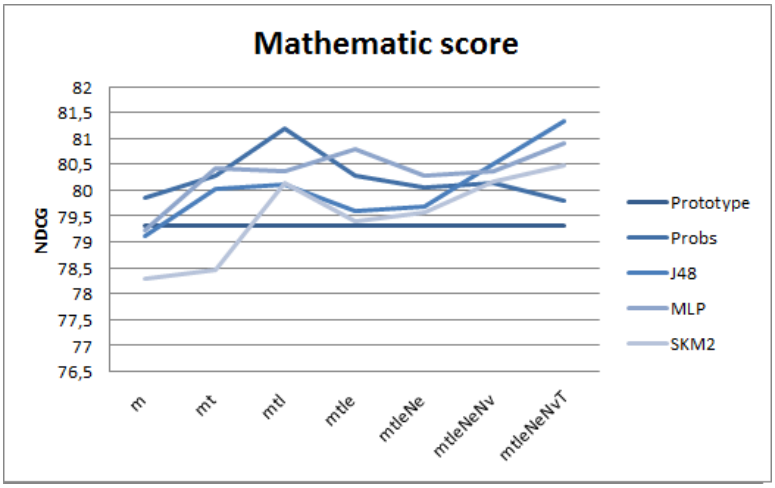


Figure 7.6: NDCG measurements for machine learning algorithm and query types using **mathematic** rules-generation approximation

In this case there are again four combinations that worsen the baseline results. The reason for this decline is the same as in the previous cases, but it helps us to determine that the new score (**Maths**) of a document not only improves the other, but worse in most cases.

7.5 Discussion

Figure 7.7 displays the numeric results (NDCG) for every combination of the algorithm. As can be seen, the results for the multimodal system using the predefined rules is

overtaken by almost every combination. The best combination is to create a model using the **'probabilities'** technique, the **'averaged-position'** ranking score and the **'mtle'** query feature that obtains a NDCG of $81,54\%$. On the contrary, the worst combination is reached using the **'SKM2'** technique, the **'mathematical'** ranking score and the **'m'** query feature that obtains a NDCG of $78,28\%$. It is also remarkable that the difference between the best and the worst combinations is only of $3,26\%$.

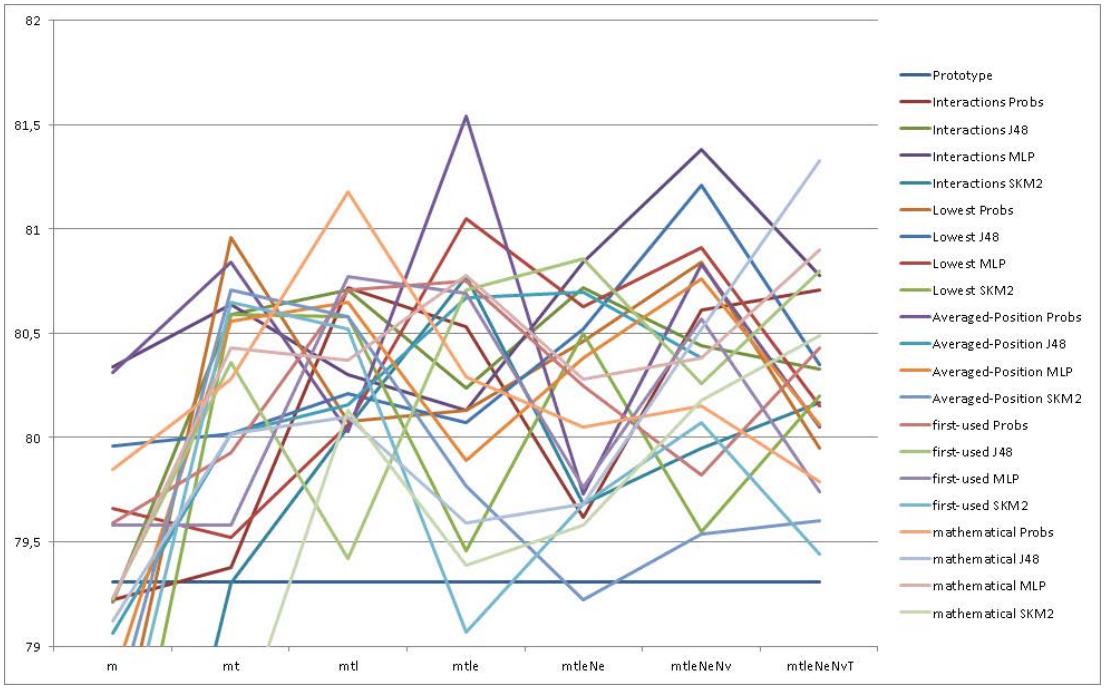


Figure 7.7: NDCG measurements for classification algorithms, query features and *REs* ranking scores.

To conclude, the results demonstrate that IR performance can be improved by using user behavior information (past interactions in this case). The numeric results also show, that IR performance improvements are limited. This is due to the fact that IR performance of every individual engine was comparable with state-of-art systems by themselves. So the combination of a set of *REs* with those performances can only get a little improvement when using their combination.

After reviewing all the results, we can conclude that none of them obtains remarkably better results than the other. It is clear that when the query features are not very specific, the results are worse than when they are more specific. This is because the

classification algorithms improved the classification task. As regards classification algorithms, there is no significant difference between them. Each is the best in any of the combinations of scores and query features. Therefore, any of these three classification algorithms could be adopted as a final option to include in the prototype. Something similar happens with measures to determine the score of the engines. Any of them would be worth us to obtain improvements in the IR, being the improvement similar in every case.

8

Development of an IMIR prototype in health domain for social media analysis

This chapter has been elaborated using Segura-Bedmar et al. [2014a], Segura-Bedmar et al. [2014b], Bedmar et al. [2015] and Martínez-Fernández et al. [2014] as sources. In addition, we also used information from an article Martínez et al. [2015] that currently (as of 09.07.2015) is under review.

Current definitions of Social Media [Kaplan and Haenlein, 2010] include several sources of user generated data, from Twitter to specialized blogs through Facebook. Users of these Web 2.0 applications share information about any subject, including issues related to their health condition. The number of people with Internet access seeking for health information through the net ranges from 70 to 75% in the U.S. Besides, 42% of them used social media to get information about health issues. Moreover, mobile technology creates an ecosystem where people are continuously accessing to the Internet and this changes the way people interact with healthcare professionals.

In this context, there is an increasing volume of digital interaction that produces a big stream of data with meaningful information that companies need to access. In

networks and forums such as *PatientsLikeMe*[117] , *DailyStrength*[118] or *Saluspot*[119] patients talk to each other about their feelings about a health problem, the way their bodies react to a given drug, how they mix different drugs to fight against some disease they have and many other issues related to their health situation. They can access health-related content as well as connect and collaborate with other patients looking for health issues.

Health insurance companies and pharmaceutical companies are very interested in not only knowing when somebody talks about a brand or topic but also identifying if they are doing it on a positive or negative way. The value of such data is not entirely established mainly because mining and analysis of social media is an emerging science.

In the healthcare scenario, there are three basic usages of user-generated data that require special attention: (a) collecting information concerning behaviors of consumers by social media analytics, (b) diffusing messages and content to a wide audience via social media channels as an addition to other media such as web sites or news portals and (c) making people and organizations aware of healthcare issues leading to a public dialogue that could be viewed by anyone.

In order to analyze this market, the heavily regulated environment around health companies and prevention of direct-to-patient interactions must be taken into account, especially in Europe. This prevents pharmaceutical companies to get involved in social networks campaigns and only half of the top 50 pharmaceutical companies in the world interact with patients through social networks. It is also worth mentioning that, outside the U.S., there are a lot of regulatory restrictions forcing pharmaceutical companies to behave in a conservative way. Nevertheless, the interest in listening patients opinions through social networks as a first step through bidirectional communication with patients is increasing.

Among all health issues, ADRs (Adverse Drug Reactions) are an important health problem due to the fact that they are the 4th cause of death in hospitalized patients [Wester et al., 2008]. Thus, the field of pharmacovigilance has received a great deal of attention due to the high and growing incidence of drug safety events [Bond and Raehl, 2006] as well as to their high associated costs [van Der Hooft et al., 2006]. Since many

[117] `https://www.patientslikeme.com/` accessed at 23/07/2015
[118] `http://www.dailystrength.org/` accessed at 23/07/2015
[119] `https://www.saluspot.com/` accessed at 23/07/2015

ADRs are not captured during clinical trials, the major medicine regulatory agencies such as the US Food and Drug Administration (FDA) require healthcare professionals to report all suspected adverse drug reactions. However, some studies have shown that ADRs are under-estimated due to the fact that they are reported by voluntary reporting systems [McClellan, 2007].

Moreover, several medicines agencies such as EMA (European Medicines Agency) and FDA have implemented web-based spontaneous reporting systems (SRS) in order for patients to report ADRs themselves. The World Health Organization (WHO) maintains the VigiBase System. These SRS have different structures and contents and almost all of them are based on voluntary reporting, except for pharmaceutical companies that are required to report suspected adverse events once they come to their attention. These companies report adverse drug events to the FDA when there is an identifiable patient, reporter and suspect drug. However, these requirements are not applied in social media

Unlike reports from healthcare professionals, patient reports often provide more detailed and explicit information about ADRs [Herxheimer et al., 2010]. The interest of having reports written by patients is that other type of information is presented and this gives a wider or complementary view of the ADR and its possible impact on the patient. Another advantage of patient reporting is that adverse effects caused by OTC (over-the-counter, medicines that are sold without prescription) medications could be analyzed. Another important contribution of spontaneous patient reporting systems is to achieve patients having a more central role in their treatments. However, despite the fact that these systems are well-established, the rate of spontaneous patient reporting is very low probably because many patients are still unaware of their existence and even may feel embarrassed when describing their symptoms or unable to describe them.

On the other side, every medicine is carefully monitored after it is placed on the market, but there are some special drugs, labeled with a black triangle , that are intensively monitored. This is due to the lack of information available about these medicines compared to others, for example because they are new in the market or there are few data about its long-term use. In this context, it is therefore essential that the safety of all medicines continues to be monitored while they are in commercial use and that suspected ADRs are reported in order to keep up to date drug packages inserts corresponding to these drugs. Presently, this pharmacovigilance work is carried

out by domain experts on a manual basis, by analyzing scientific literature as well as clinical trials documents and spontaneous reports.

Harpaz et al. [2012] remarked that new methods that integrate data extracted from SRS narratives and knowledge extracted from experimental preclinical discovery drugs sources are required. Furthermore, patient-generated content concerns also discussions about treatments and opinions about drugs that could lead to valuable knowledge. Patients use Social Media to self-report adverse drug events three times more often than reporting to FDA [Freifeld et al., 2014] and 90% is the estimated rate of ADRs that patients dont report [23]. Thus, the main hypothesis of this article is that health-related social media can be used as a complementary data source to spontaneous reporting systems as well as to help pharmacovigilance to report about the incorrect use of drugs, that is, monitoring of abuse and misuse of medicinal products, for instance by people that have problems to understand medical language.

8.1 Trendminer project

TrendMiner is a research and development (R&D) project co-funded by the European Commission (287863) in the Seventh Framework Programme. It started at 2011-11-01 with a duration of 36 months.

The goal of this project is to deliver innovative, portable open-source real-time methods for cross-lingual mining and summarisation of large-scale stream media. TrendMiner will achieve this through an inter-disciplinary approach, combining deep linguistic methods from text processing, knowledge-based reasoning from web science, machine learning, economics, and political science. No expensive human annotated data will be required due to our use of time-series data (e.g. financial markets, political polls) as a proxy. A key novelty will be weakly supervised machine learning algorithms for automatic discovery of new trends and correlations. Scalability and affordability will be addressed through a cloud-based infrastructure for real-time text mining from stream media.

Results are validated in high-profile case studies: financial decision support (with analysts, traders, regulators, and economists), political analysis and monitoring (with politicians, economists, and political journalists), detection of psychosocial states and

social information, and detection of discussions on medicine and drug effects in social media.

The results aimed in TrendMiner are: novel models and approaches for combining multi-lingual text processing, extra-linguistic knowledge, and time-series machine learning models, open-source algorithms for real-time analysis and summarisation of multilingual media streams, a cloud-based platform for real-time stream media and two demonstrated deployments in financial decision support and political science.

The partners taken part in the project are: Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (Germany), The University of Sheffield (United Kingdom), Ontotext AD (Bulgaria), University of Southampton (United Kingdom), Internet Memory Research (France), Eurokleis S.R.L. (Italy), Sora Ogris & Hofinger GmbH (Austria), Hardik Fintrade Pvt Ltd. (India), Universidad Carlos III of Madrid (since 01.11.2913), Department of Corpus Linguistics of the Hungarian Academy of Sciences (since 01.11.2013), Institute of Computer Science Polish Academy of Sciences (since 01.11.2013) and DAEDALUS - DATA, DECISIONS AND LANGUAGE, S. A. (since 01.11.2013).

8.2 System to monitoring health social media: drugs, effects and relations extraction and retrieval

Monitoring social media health information is done from a prototype based on the formal model (see chapter 3). This prototype is responsible for obtaining information from social media, analyzing messages linguistically, storing the information properly (indexing) and displaying messages to users.

The health-domain prototype (shown in Figure 8.1) is characterized by three parts. The first one is the semantic resources that help the analysis of the documents and the grouping of information in the GUI. The second part is the annotation pipeline, which is responsible for processing the documents to be added to the storage system (datawarehouse), and the last part is the interactive multimodal information retrieval (IMIR) system of health-domain. Within three parts, our work has focused mainly on the development of annotation pipeline. The development of the other two parts has been shared by us but has been mainly attributed to other project partners. These three parts are described below.

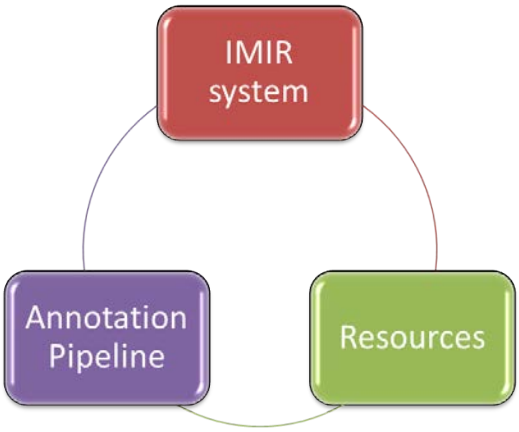


Figure 8.1: Health monitoring system schema

8.3 Health resources: Drugs, Diseases and Effects

Several semantic resources (shown in figure 8.2) have been integrated in the system. Each resource is intended to detect a different type of named entity (or relations). These semantic resources are explained below.

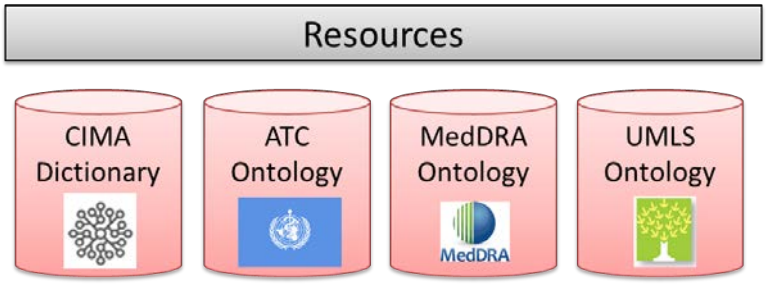


Figure 8.2: Resume of the semantic resources used in the health monitor system

8.3.1 CIMA

CIMA is a resource provided and maintained by The Spanish Agency for Medications and Healthcare Products (AEMPS). It is an application which includes all authorized drugs in Spain.

From CIMA files, 16,418 drugs, 2,228 active substances and 3,659 brand drugs were

obtained. Additionally, 4,817 drug related terms were obtained from **Vademecum**[120] (a guide of pharmaceutical products that includes over 18,200 drugs) and from **MedlinePlus**[121], the National Institutes of Health's (NIH) website intended for patients. These terms compose the gazetteer **DrugsGaz**.

In order to relate brand names and active substances we use the Anatomical Therapeutic Chemical (ATC)[122] classification system that consists on a set of alphanumeric codes developed by the WHO for the classification of drugs and other medical products organized in 5 levels. Figure 8.3 shows an example of the five levels of ATC. All this knowledge is related to in a dictionary called **drugsATC**. Each entry corresponds to a drug (brand name) followed by those active substances that compose it as aliases.

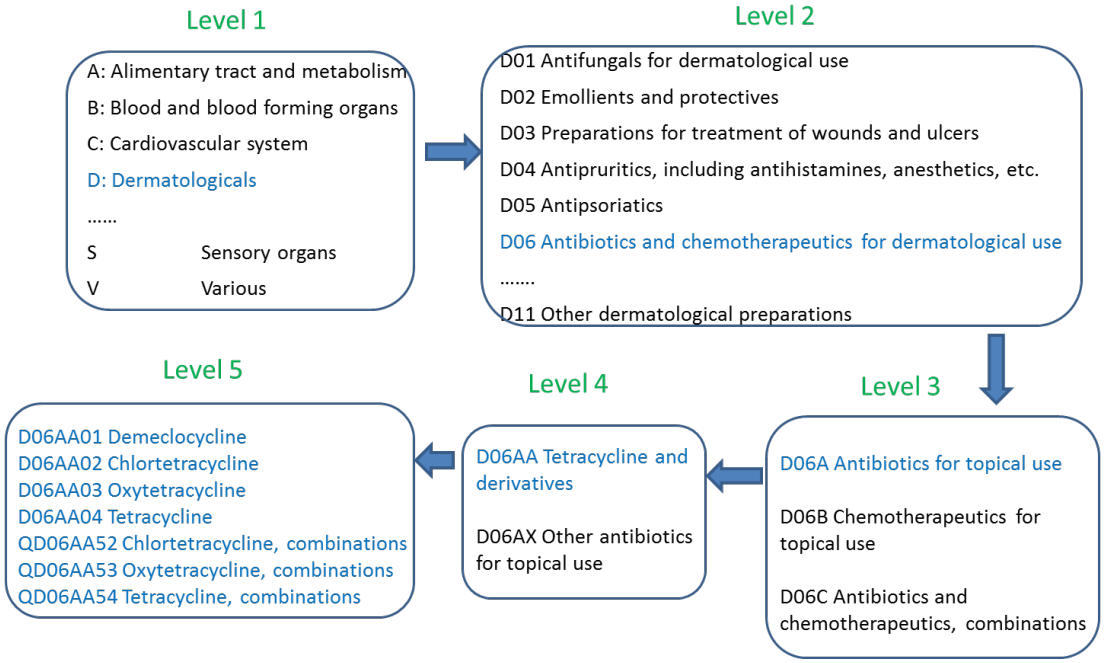


Figure 8.3: Example of ATC system structure

A complete description of CIMA and the resources generated from it (**DrugsGaz** gazetteer and **drugATC** dictionary) can be found in Segura-Bedmar et al. [2014b].

[120] `http://www.vademecum.es/` accessed at 23/07/2015
[121] `http://www.nlm.nih.gov/medlineplus/spanish/` accessed at 23/07/2015
[122] `http://www.atccode.com/` accessed at 23/07/2015

8.3.2 MedDRA

MedDRA is the adverse event classification dictionary approved by the International Conference on Harmonization of Technical Requirements for Registration of Pharmaceuticals for Human Use (ICH), and therefore a very reliable resource for the adverse events.

MedDRA supports ten languages and is composed of a five levels hierarchy which goes from more general to very specific. The two lower levels from MedDRA PT (Preferred Terms) and LLT (Lowest Level Terms) were extracted to implement the **adrsMedDRA** dictionary for ADRs detection. The information we obtained from this resource is: 13,245 PT adverse effects and 35,259 LLT adverse effects.

A complete description of MedDRA and the **adrsMedDRA** dictionary can be found in Segura-Bedmar et al. [2014b].

8.3.3 UMLS-SNOMED CT

UMLS, developed by the National Library of Medicine (NLM), is a comprehensive list of medical terms mainly focused on developing computer systems suitable for understanding the specific vocabulary which is normally used in biomedicine and health care literature. One of the resources integrated in UMLS is SNOMED CT (Systematized Nomenclature of Medicine - Clinical Terms), a terminology accessible in Spanish whose content consists of concepts, descriptions and relationships to represent information and clinic knowledge.

UMLS is structured in several semantic categories (substances, organisms, health care activity, etc.). Three of these categories (Diseases or syndrome, Mental or Behavioral Dysfunction and Neoplastic process) have been chosen in order to create the **diseasesUMLS** dictionary for diseases and symptoms. The information we obtained from UMLS Database is 42,548 main diseases and 23,677 diseases synonyms.

A complete description of UMLS-SNOMED CT and the **diseasesUMLS** dictionary can be found in Segura-Bedmar et al. [2014b].

8.3.4 The SpanishDrugEffectDB Database

The last resource is a database that stores relations between drugs and effects.

Although there are several English databases such as SIDER[123] or MedEffect[124] with information about drugs and their side effects, none of them are available in Spanish. Moreover, these resources do not include drug indications. Furthermore, there are other initiatives to build knowledge bases in English with ADRs from drug package inserts that can be used to assess ADRs such as Boyce et al. [2014]. In this work **SpanishDrugEffectDB** [Segura-Bedmar et al., 2014a], with information about drugs, their drug indications as well as their adverse drug reactions in Spanish has been integrated.

A complete description of **SpanishDrugEffectDB** database can be found in Segura-Bedmar et al. [2014b].

8.4 Offline health annotation pipeline

This pipeline is our main contribution. We were responsible (inside the whole Trendminer project) for the annotation process. A sequential annotation pipeline (implemented using GATE[125]) is in charge of annotation and post-filtering tasks (see figure 8.4). The system manages the semantic annotation of the text documents (user comments and tweets). Finally, it stores the response given by the pipeline back to the Elasticsearch[126] data warehouse (see section 8.5.4).

The core semantic technology used in this pipeline is the Meaningcloud[127] commercial tool. It offers several semantic APIs in SaaS (Software as a Service) mode which can process all kinds of unstructured multimedia content to extract elements of meaning (topics, facts, opinions, relationships, etc.).

The annotation pipeline is composed of six stages (shown in figure 8.4):

1. Language Identification: stage that discards every text that is not written in Spanish. The Twitter API was asked for texts only in Spanish, but since this may sometimes fail, another filtering step is performed while analyzing the document. The identification is made by the Meaningcloud Language Identification API, which uses statistical techniques based on n-grams.

[123] `http://sideeffects.embl.de/` accessed at 23/07/2015
[124] `http://www.hc-sc.gc.ca/dhp-mps/medeff/index-eng.php` accessed at 23/07/2015
[125] `https://gate.ac.uk/` accessed at 23/07/2015
[126] `https://www.elastic.co/products/elasticsearch` accessed at 23/07/2015
[127] `https://www.meaningcloud.com/es/` accessed at 23/07/2015

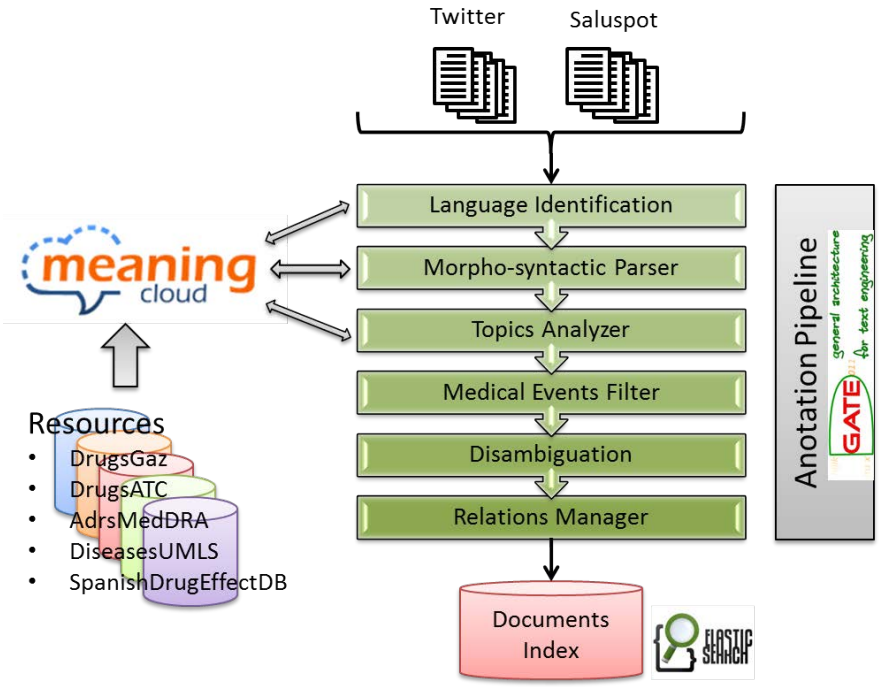


Figure 8.4: Components and processing flow of the annotation pipeline

2. Morpho-syntactic Parsing: it is performed by the Meaningcloud Lemmatization, Part-of-speech (PoS) Tagging and Parsing API which follows a dictionary-based approach to morpho-syntactically analyze the texts. This step is of great importance in order to perform the disambiguation step that comes later, due to the high ambiguity that exists in medical texts. The output of the stage is composed by the morpho-syntactical annotations.

3. Topics Analyzer: several health-related dictionaries were created to NER (*DrugsGaz*, *DrugsATC*, *AdrsMedDRA*, *DiseasesUMLS* and *SpanishDrugEffectDB*) and integrated in the Meaningcloud Topics Extraction API (see section 8.3). In order to ease the use of the Meaningcloud APIs inside of the well-known and extensively used GATE Platform, a plug-in was created and made public through a Plugin repository which is available from the Meaningcloud website. Topics analyzer works in a fuzzy way in order to detect drugs with misspelled errors. The output of this stage are the medical domain annotations.

4. Medical Events Filter: it filters all the entities that have been annotated by the Topics Analyzer and which are not from the medical domain. Only drug, adverse effect and disease entities are kept in the system. For example, in the sentence '*I hear Rolling Stones when I suffer headache*' two named entities are annotated (underlined). 'Rollign Stones' is annotated as a named entity of the music domain and 'headache' in the health domain. When applying the filters, every named entity annotation of non-health domains is deleted, so only 'headache is annotated as named entity.

5. Disambiguation: A set of rules that uses linguistic features like the morpho-syntactic information provided by the parser, together with co-occurrence information of drugs and diseases, are used to filter out terms that are not likely to be mentions of medical events. If we look at the example shown in figure 8.5, *'motivan'* is a drug name in the dictionary (it is an antidepressant whose active substance is paroxetina). However, in this case the morpho-syntactic parser detects that it plays as a verb (*motivar*) in the sentence and the parser does not tag it as a drug entity.

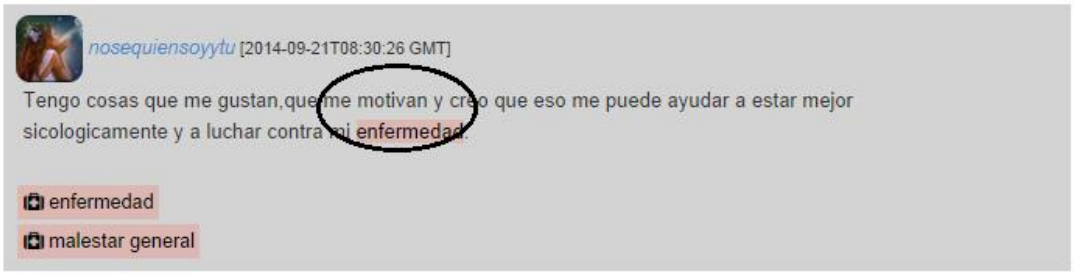


Figure 8.5: Example of tweet annotated after the post-filtering stage where *'motivan'* is detected as verb and it is not tagged

6. Relations Manager: this component annotates three types of relations between drugs and diseases or effects, classifying them into (1) adverse effects, (2) indications or (3) pairs that hold a possible relationship. The two first classifications are relations that were extracted from the **SpanishDrugEffectDB** database (see section 8.3.4), that has been built from several websites containing drug package inserts as it is explained in section 8.3. In contrast, the latter group has been

created to point out possible un-catalogued or unknown relations that may be discovered due to situations of high recurrence pointed out by the system. For instance, Figure 8.6 shows an adverse effect between drug *'Taxotere'* (Taxotere) and effect *'eritema cutáneo'* (cutaneous erythema).

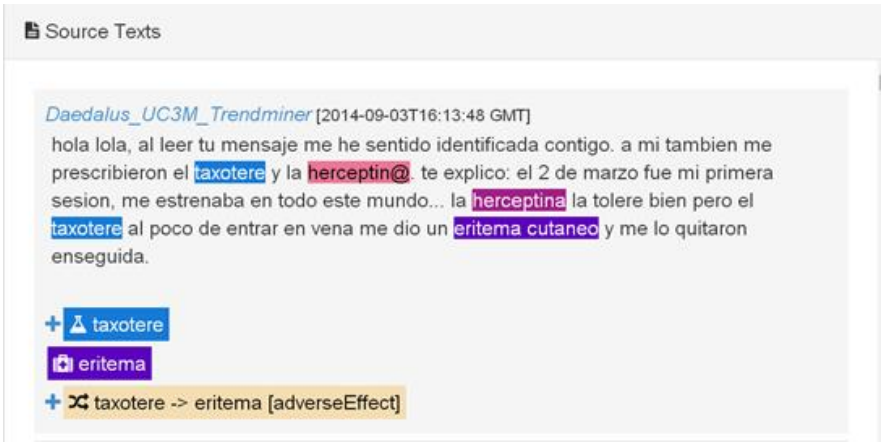


Figure 8.6: Example of a comment tagged with drugs, effects and relations

8.5 Health IMIR System

The third part of the social media monitor system is the retrieval system that is in charge of requesting the datawarehouse generated by the pipeline. There are seven parts composing the retrieval system: user generated health information, knowledge bases, query, retrieval engines, sources handler, results' combination module and graphical user interface. The prototype is available online at `http://trendminer.daedalus.es/` (at 23/07/2015).

The whole architecture of the health IMIR system and the processing flow are depicted in figure 8.7. This flow begins in the graphical user interface when a user posts a query. The user also selects the retrieval engine that is requested. Then, the query and the user-selected engine are sent to the handler. After requesting the selected RE, the handler returns the results' set to the graphical interface. There is no results' combination module because only one *RE* is requested at each search. For more details see section 8.5.6.

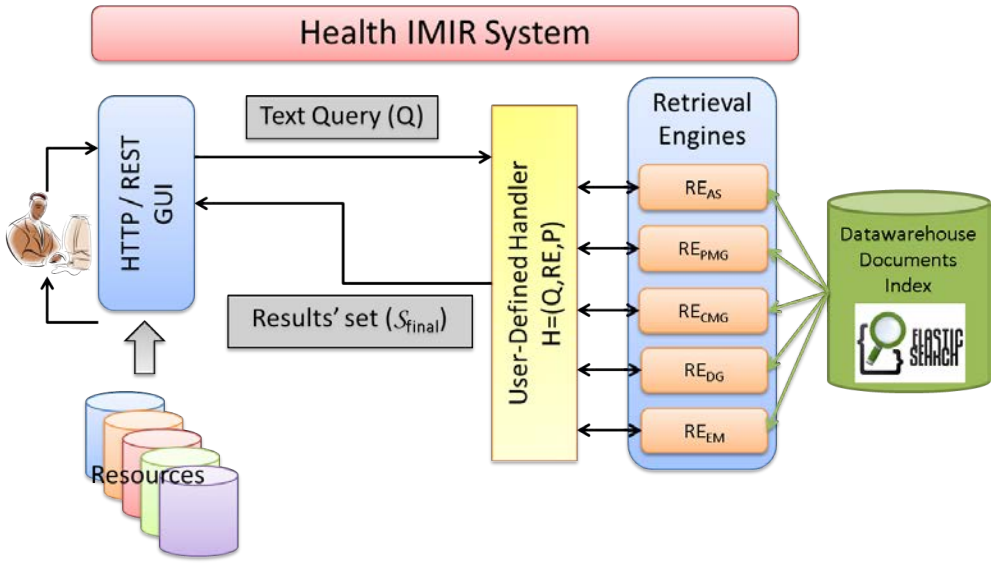


Figure 8.7: Health-domain IMIR system architecture showing its main components. It encompasses five retrieval engines (see section 8.5.4) that are: RE_{AS} is the active substance search engine, RE_{PMG} is the pharmacological main group search engine, RE_{CMG} is the chemical main group search engine, RE_{DG} is the downwards grouping search engine and RE_{EM} is exact match search engine

8.5.1 Health information

The first component defined by the model is multimodal information (see section 3.2.1). In this case two sources of user generated data related to health issues are used: Twitter and Saluspot. From Twitter, only tweets corresponding to certain filters are collected. Concretely, tweets that contain specific keywords like drug or disease names and are written in Spanish (the prototype currently collects tweets containing antidepressants and related drugs). The second data source is Saluspot, a Spanish website that allows its users to address free of charge and anonymously their doubts and information needs about health, lifestyle and drugs to thousands of registered doctors. Once a question is posted any of the registered, accredited doctors can answer and even multiple answers are possible. The system continuously evolves, but in principle, each question contains information about the users gender and age, the date of posting and one or more answers together with the identity of the doctor who answered and a reliability measure based on the number of doctors who accepted to tackle this particular question.

A complete description of the gathering process of both collections can be found in Segura-Bedmar et al. [2014b].

Multimodal information is sorted into two collections, Twitter ($C_{twitter}$) and Saluspot ($C_{saluspot}$), formally defined in equation 8.1.

$$\mathcal{C}_{t-s} = \{C_{twitter}, C_{saluspot}\} = \{C_t, C_s\} \tag{8.1}$$

Each collection is composed of a set of documents where 41.991 is the number of documents in the **Saluspot** collection and 2.758.371 is the number of documents in the **Twitter** collection. The formal definition of each document is shown in equation 8.2.

$$\begin{aligned} C_t &= \{D_{t,1}, D_{t,2} \dots D_{t,2.758.371}\} \\ C_s &= \{D_{s,1}, D_{s,2} \dots D_{s,41.991}\} \end{aligned} \tag{8.2}$$

Each document (see equation 8.3) consists of a set of elements where P represents the number of elements of document $D_{i,j}$ and each element $d_{i,j,k}$ is a text element, i.e., $\mathcal{M}(d_{i,j,k}) = txt \ \forall i, j, k$.

$$D_{i,j} = \{d_{i,j,1}, d_{i,j,2}, \dots d_{i,j,P}\} \tag{8.3}$$

8.5.2 Health knowledge bases

Unlike knowledge systems described in the formal model, in this case knowledge systems do not relate semantically documents between them (multimedia relations) but they are only used to annotate documents with the concepts of knowledge bases (semantic relations). The knowledge bases used in this prototype have been described in section 8.3.

8.5.3 Text Query

In the health domain, it is difficult to find a scenario where the query has to include multimedia elements, so this prototype has been implemented for accepting queries in one mode, text. On the contrary, the queries can be classified into two types: raw text and concepts. The raw text queries are used for exact-match searches (see section 8.5.4) while the concept queries are concepts extracted from the knowledge bases and are used for the other search modalities (active substance, chemical main group, pharmacological

main group and downward grouping) that are given in an auto-complete search box in the GUI. The query is defined as a set of text elements

$$Q = \{q_1, q_2, \ldots, q_K\} \tag{8.4}$$

where:

- K is the number of elements of query Q.
- Each element q_k is a text element ($\mathcal{M}(q_k) = txt$).

8.5.4 Retrieval engine

A data warehouse based on *Elasticsearch*[128] is responsible for efficiently storing the high volume of real-time data from social networks that the system manages, as well as for providing advance search functionality that allow the visualization module to generate complex analytics. *Elasticsearch* is a flexible, powerful, open source, distributed and real-time search and analytics engine. Some of the key factors that made us take the decision of choosing this architecture are its distributed capabilities and the fact that it can easily and horizontally scale when the system growth starts affecting performance. Furthermore, *Elasticsearch* runs on top of *Apache Lucene*, so that it offers quite complex search capabilities, high-performance and is trustworthy, due to its well-known reliability.

The index has currently a size of more than 2 million documents that comprises about 1.7GB of data. The system has been collecting data starting from 17th July 2014 and is up right now. Some statistics from the documents stored in the Datawarehouse, at the moment of writing this are: (i) 41,991 Saluspot posts annotated with 1,864 mentions of unique drugs, 1,581 of unique diseases and 2,089 of unique adverse effect mentions. It also contains 18,397 unique relations (1987 adverse effects, 459 indications and 15,951 uncategorized relations); and (ii) 380,000 tweets containing a total of 564 unique drugs, 73 unique diseases, 170 unique adverse effects and 587 unique relations (97 adverse effects, 29 indications and 461 possible relations). The overall numbers for the whole dataset are 1965 unique drugs, 1591 unique diseases, 2103 unique adverse

[128] `http://www.elasticsearch.org/` accessed at 23/07/2015

effects and 18820 unique relations (2031 unique adverse effects, 465 unique indications and 16324 unique uncategorized relations).

A complete description of the datawarehouse can be found in Martínez-Fernández et al. [2014].

The information contained in this datawarehouse is retrieved through five different retrieval engines. Each retrieval engine requests a different type of information of the data warehouse. These engines are:

1. **Active Substance**: $RE_{AS} = [\mathcal{C}_{t-s}, Q, \mathcal{P}_{AS}]$ where $\mathcal{P}_{AS}$ is based on the ATC code of the drugs of level 5 of the ATC structure and is based on grouping mentions that share the same group at each corresponding level. If a user searches for 'Tetracycline (D06AA04)', then every drug containing the same active substance is retrieved.

2. **Pharmacological Main Group**: $RE_{PMG} = [\mathcal{C}_{t-s}, Q, \mathcal{P}_{PMG}]$ where $\mathcal{P}_{PMG}$ is based on the ATC code of the drugs of level 3 of the ATC structure and is based on grouping mentions that share the same group at each corresponding level. If a user searches for 'Antibiotics for topical use (D06A)', then every drug belonging to the same active pharmacological group is retrieved: 'Oxytetracycline', 'Chlortetracycline', 'Tetracycline', etc.

3. **Chemical Main Group**: $RE_{CMG} = [\mathcal{C}_{t-s}, Q, \mathcal{P}_{CMG}]$ where $\mathcal{P}_{CMG}$ is based on the ATC code of the drugs of level 4 of the ATC structure and is based on grouping mentions that share the same group at each corresponding level. If a user searches for 'Tetracycline and derivatives (D06AA)', then every drug belonging to the same active pharmacological group is retrieved: 'Oxytetracycline', 'Chlortetracycline', 'Tetracycline', etc.

4. **Downwards Grouping**: $RE_{DG} = [\mathcal{C}_{t-s}, Q, \mathcal{P}_{DG}]$ where $\mathcal{P}_{DG}$ gets the element that defines the search (in the ATC structure tree) and groups together every element below this element.

5. **Exact Match**: $RE_{EM} = [\mathcal{C}_{t-s}, Q, \mathcal{P}_{EM}]$ where $\mathcal{P}_{EM}$ looks for specific mentions of the exact terms regardless of the ATC code of the mentions.

8.5.5 Handler

As in any system requesting more than one engine, it must have a Handler that decides which engines to request with each query. The handling strategy execution is a parallel execution following a strategy of exclusion, i.e., only one of the available retrieval engines is requested. The selection of this engine is done by the user. Therefore, the strategy is a manual source selection and its functionality moves directly to the graphical user interface.

The handling strategy is defines as a triplet in equation 3.10. In the health handler, the definition is particularized as

$$\mathcal{H} = [\mathcal{E}t - s, Q, \Xi_{health}] \tag{8.5}$$

where:

- $\mathcal{E}_{t-s}$ represents a set of available REs: $\mathcal{E}_{t-s}$ =$\{RE_{AS}$, RE_{PMG}, RE_{CMG}, RE_{DG}, $RE_{EM}\}$.
- Q represents the input query.
- Ξ_{health} is the handling strategy.

The functionality of the handling strategy (Ξ_{health}) (see equation 3.11) of this prototype is particularized for being defined as a selection strategy that selects a unique RE (see equation 8.6).

$$\mathcal{E}'_{t-s} = \{RE_{unique}\} \tag{8.6}$$

where RE_{unique} represents the selected retrieval engine and $unique \in \{AS, PMG, CMG, DG, EM\}$.

8.5.6 Results' Combination and Aggregation

The combination of results is divided into two parts: (1) the first part is the combination that is done in the datawarehouse (joint index) as documents, although they come from different sources and have different formats, are indexed in the same way and stored together; and (2) the second part is the combination that is made while displaying

the results in the graphical interface. The interface has three ways to combine (or aggregate) the results which are described below:

1. **Lists** shows a set of documents sorted by creation date of the document. This form of visualization does not combine results, but it displays them chronologically. Figure 8.8 shows a screenshot of the results list ordered chronologically.

Figure 8.8: Visualization of a list of tweets resulting from a search

2. **Concepts cloud** displays a set of concepts of the knowledge bases with which the results are annotated. As can be seen, the size of every concept depends on the number of results annotated with this concept. An example of concepts cloud is shown in figure 8.9.

3. **Temporal aggregation bar graphs** displays bar graphs that aggregate the number of mentions of discovered relations of different kinds for the texts that match the query. An example is given in figure 8.14.

8.5.7 Graphical user interface: Dashboard

Finally, the graphical user interface performs the analytics in order to display meaningful relations, patterns of co-occurrence, and other data insights to the final user. These

Figure 8.9: Concepts cloud of the graphical interface showing Drugs annotated in the results

visualization modes are related to the five search modes (see section 8.5.4). The prototype allows viewing the annotated source texts that match a specific search, focusing on their drug and disease mentions and showing the discovered relations, like shown in the example of figure 8.8.

In order to enhance usability, the search box is designed to display every possible term in our vocabulary. The resources used to build the different dictionaries are also compiled and indexed into another *Elasticsearch* index, using an n-gram analyzer at index time (using from 2 to 20 grams for each word indexed). In contrast, at search time standard tokenizer, lower case token and stopwords filters are applied. By doing this, the system quickly responds to the user of the system with the hundred most likely terms that match the input provided by the user. Another advantage of this approach is that it allows looking for both the canonical form and any of the synonyms or alias that the term has in our database. See Figure 8.10 for more details.

In figure 8.11 individual bar graphs aggregating the number of mentions of discovered relations of different kinds for the texts that match the query are displayed.

Examples of co-occurrences between drugs (drug-drug) and between drugs and diseases (drug-disease) are given in figure 8.12.

In addition for each active substance and brand name, their classification in the ATC tree is also showed as well as links to external sources (see Figure 8.13). In order

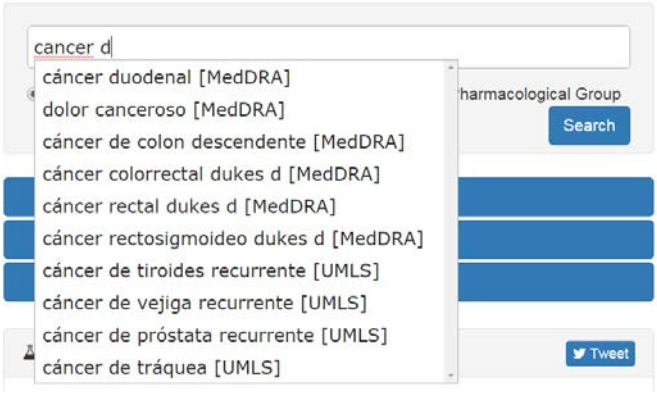


Figure 8.10: Example showing search options using cáncer (cancer) query

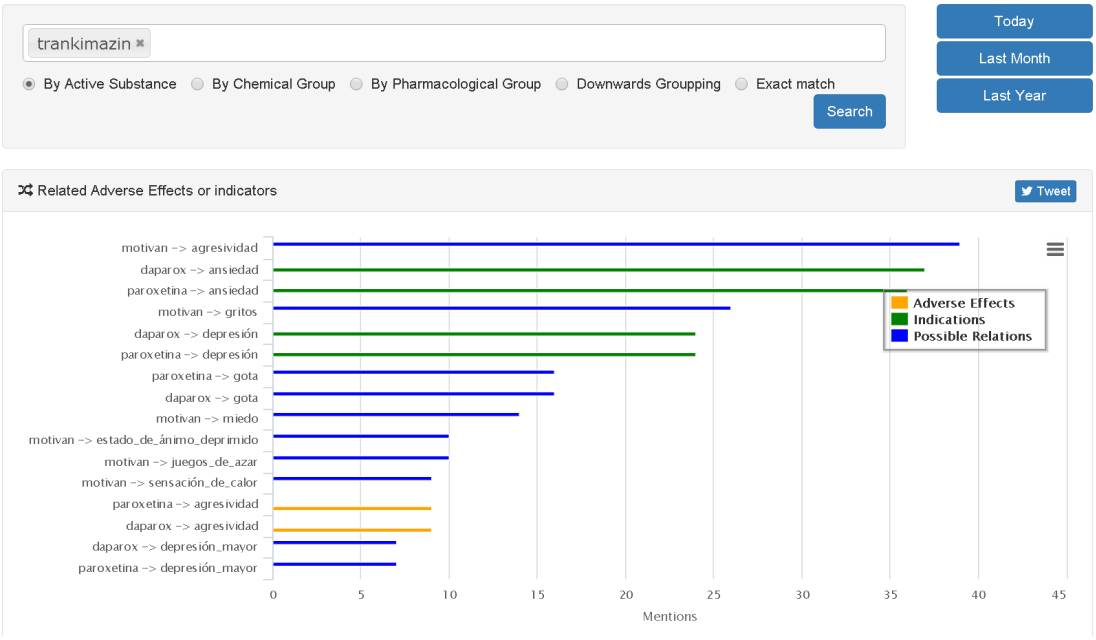


Figure 8.11: Graph showing aggregated data about effects related to drug Trankimazin (indications, ADRs and possible relations)

to provide cross-lingual capabilities, ATC classification is displaying in other languages such as German and English.

Finally, the system also presents information about the evolution of mentions through a timeline graph with different granularity (months, weeks, days) like the one shown in Figure 8.14. All graphs have been developed using Highcharts.

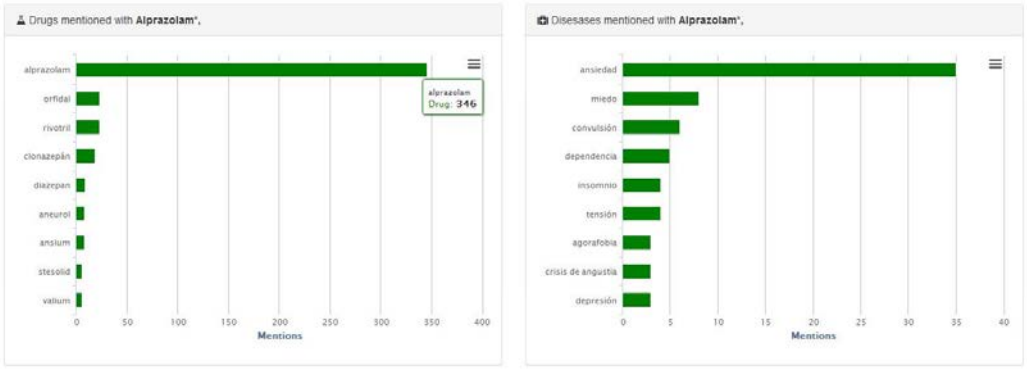


Figure 8.12: Graph showing co-occurrence aggregated data for Alprazolam active substance

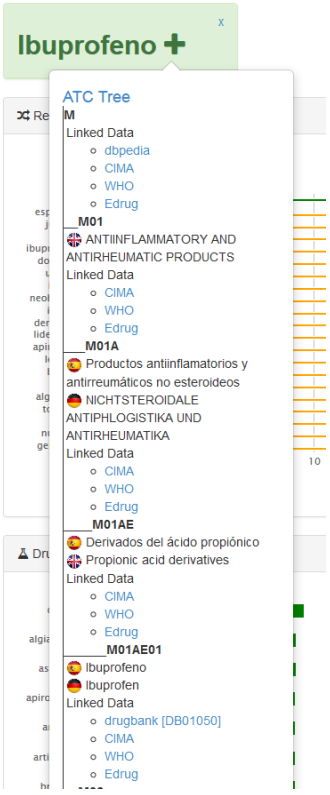


Figure 8.13: Display of multilingual ibuprofeno ATC tree

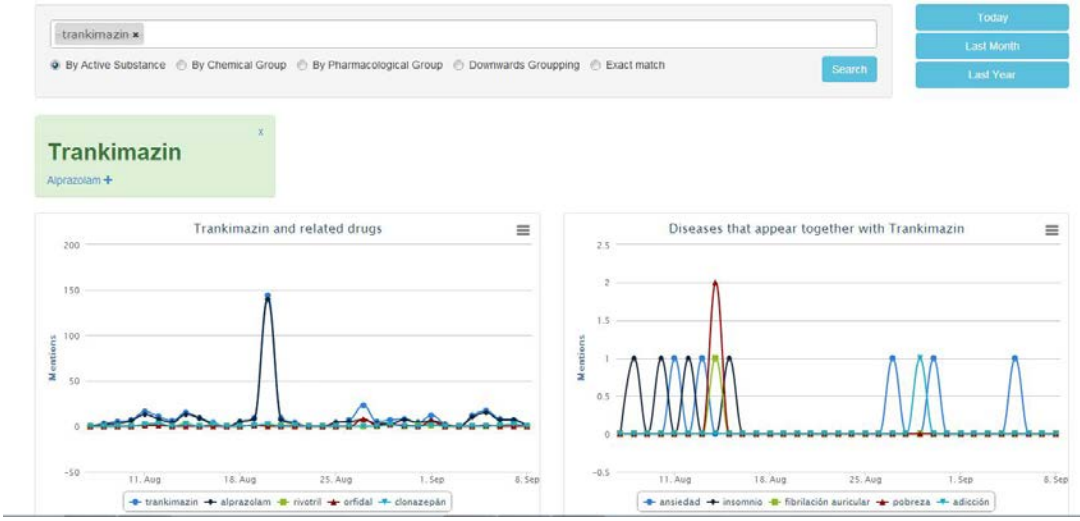


Figure 8.14: Graph showing time based evolution of entity mentions for Trankimazin query grouped by active substance

8.6 Some experiments evaluating NER and relation extraction

In order to evaluate the linguistic processor we have used a corpus extracted from ForumClínic[129], an interactive web page intended for patients to increase their degree of autonomy with respect to health issues, using the opportunities given by the newest Web technologies. Its target is to improve citizen's knowledge on health, diseases and their causes, as well as the efficiency and safety of the preventive treatments and medicines, so that they can get involved with the clinical decisions which attain them.

ForumClínic users are from all over the world, but a significant data is the fact that 46% of the webpage visits come from Spanish speaking countries in America. In total, the number of a million users was reached in 2011, and it maintains a steady increase since it was created, in 2007.

To accomplish the evaluation we have used the SpanishADR corpus [Segura-Bedmar et al., 2014b] which consists of 400 user messages collected from ForumClínic. The size of the corpus is 26,519 tokens, whereas each message contains an average of 3.15 annotations (0.48 drugs, 1.42 effects and 1.25 relations). Moreover, it contains 189

[129] `http://www.forumclinic.org/` accessed at 23/07/2015

drug annotations, 568 effect annotations and 164 drug-effect relations (the extension of SpanishADR corpus with drug-effect annotations is described in [Segura-Bedmar et al., 2014a]).

Metrics used are precision (P), recall (R) and F-measure. P, R and F-measure are calculated according to two different criteria: the strict matching considers as correct every response where type entity and the spans are identical and the lenient matching considers every partially correct response as correct, i.e. the entity type is correct and the spans are overlapping but not identical.

Concerning NER, table 8.1 shows P, R and F-measure evaluating drug recognition. The main source of false negatives for drugs seems to be the abbreviations for drug families. For instance, *'benzodiacepinas'* (benzodiazepines) is commonly used as benzos, which is not included in our dictionary. An interesting source of errors to point out is the use of acronyms referring to a combination of two or more drugs. For instance, FEC is a combination of Fluorouracil, Epirubicin and Cyclophosphamide, three chemotherapy drugs used to treat breast cancer. Related to false positives some drug names such as *'alcohol'* (alcohol) or *'oxígeno'* (oxygen) can take a meaning different than the one of pharmaceutical substance. Another important cause of false positives is due to the use of drug family names as adjectives that specify an effect. This is the case of *'sedante'* (sedative) or *'antidepresivo'* (antidepressant), which can refer to a family of drugs, but also to the definition of an effect or disorder caused by a drug (sedative effects)

Drugs	R	P	F-Measure
strict	0,68	0,85	0,76
lenient	0,68	0,85	0,76

Table 8.1: Evaluation measures in drug recognition.

Table 8.2 shows P, R and F-measure evaluating effect recognition. The major source of false negatives was the use of colloquial and lay expressions to describe an effect. Patients used expressions such as *'tengo la cabeza como un bombo'* (my head is ringing) or *'estoy destrozado'* (I am destroyed) in order to express how they felt. These expressions are not included in our dictionary. A possible solution could be to create a lexicon containing these colloquial expressions. The second highest source

of false negatives for effects was due to the different lexical variations of the same effect. For instance, *'estrés'* (stress) is a term included in our dictionary, but its lexical variations, like for example *'estresado'* (stressed), *'estresante'* (stressful), *'me estreso'* (I get stressed), *'me estresa'* (it makes me feel stressed) are not, and therefore they were not detected by our system. Nominalization may be used to identify all the possible lexical variations of a same effect. The third largest source of false negatives was spelling mistakes. We can see an example with *'hurticaria'*, which is an incorrect way of writing *'urticaria'* (urticaria). Many users have great difficulty in spelling unusual and complex technical terms.

Effects	R	P	F-Measure
strict	0,43	0,75	0,54
lenient	0,47	0,83	0,6

Table 8.2: Evaluation measures in effect recognition.

False positives for adverse events were mainly due to the lack of ambiguity resolution. Some medical events receive the name of a common Spanish word, as it happens with Zona (Herpes zoster). Also acronyms used for long-named adverse events sometimes match with common words. For example, *'Infección Respiratoria Altas'* (Upper Respiratory Tract Infection) acronym, *'IRA'*, has different meanings in Spanish (past form of the verb to go or anger among others). Furthermore, some effects such as *'anestesia'* (anesthesia) share the name with the drug which drives patients to that state.

Table 8.3 shows P, R and F-measure evaluating relation extraction taking into account drug-effect pairs annotated in the corpus (the objective is to evaluate relation extraction task regardless of NER task). Regarding the false positives, a cause of error is SpanishDrugEffectDB could include incorrect relations due to the fact that it was automatically obtained and it has not been manually revised. Another source of errors is the lack of context resolution. This means that, despite correctly detecting a drug and an effect (according to the drug package information), the context of the text did not fulfill the requirements to properly consider it a relation. Moreover, the lack of co-reference resolution introduces another important source of error for false positives;

terms such as *'enfermedad', 'efecto', 'tratamiento'* and other have to be solved. An interesting source of errors is the lack of negation resolution, which means that despite the fact that the user specifies that he/she did not experience an effect after taking a drug, the system annotates the relation. Finally, the complex sentences (coordinated and subordinated sentences) in a comment may mislead the system into annotating a relation which is not correct.

		SpanishDrugEffectDB			Coocurrences		
Windowsize		R	P	F-Measure	R	P	F-Measure
30	strict	0,08	0,57	0,14	0,63	0,44	0,52
30	lenient	0,13	0,96	0,24	0,88	0,61	0,72
100	strict	0,1	0,34	0,16	0,74	0,26	0,38
100	lenient	0,23	0,74	0,35	0,99	0,34	0,51
250	strict	0,12	0,32	0,17	0,17	0,75	0,33
250	lenient	0,24	0,67	0,36	1	0,29	0,45

Table 8.3: Evaluation measures in relation extraction (over drug-effect annotated pairs in Goldstandard corpus).

Finally, concerning false negatives table 8.3 shows that a great number of drug-effect pairs appearing in the corpus are not covered by the SpanishDrugEffectDB (recall is very low), that is, this database does not include all drugs effects. Therefore, the corpus has only 164 relations and it is difficult to conclude about the database coverage.

8.7 Discussion

A domain as health social media streams retrieval poses several challenges and requires innovative ICT (Information and Communications Technology) products and services such as scalable NLP technology. A fully functional prototype, based on the formal model presented in chapter 3, for annotating and retrieving information from social media streams in the health domain has been implemented. This prototype accepts query in one mode and returns monomodal (text) results from two collections of health

documents (Saluspot and Twitter). Five retrieval engines are implemented to request a combined index where the collections have been annotated and indexed. It uses a selection handler that requests only one retrieval engine at a time. The results are directly combined in the joint index. Nevertheless, the graphical interface has implemented aggregation mechanisms that help to the visualization of aggregated information about the returned results. This prototype constitute a first approach for us to build a more complex system driven to the health sector.

An evaluation has been also performed using a corpus annotated with drug-effect pairs [Segura-Bedmar et al., 2014b] and an analysis of errors has been done with the aim of identifying future improvements. One issue that requires special attention is to manage patient oriented vocabulary. Patients do not report about their treatments using clinician terminology. Consumer Health Vocabulary [Smith and Stavri, 2005] is a terminology for English language that contains lay terms but Spanish requires a similar resource that could be (semi)automatically built using NLP.

9

Final Remarks

9.1 Conclusions

Current society is characterized by a constant technological revolution, where the generation and consumption of information is reaching huge levels. Devices and formats are very diverse and move away from traditional modes. Retrieval methods do not remain constant and become dependent on the device used to query (smartphones, tablets, PCs, etc.), what is being queried and who is querying. Users should find all the information they need easily and without having to request different sources. In addition, users also want to search in a more complex way with different modes, so new features have to be contemplated.

The growing presence of multimedia online content (internet, corporate intranets, etc.) motivates the problem that this thesis covers: users need to make multimodal requests to multimodal search engines to access bigger and bigger amounts of information in plenty of different formats (such as video, text, audio, images, graphics, etc.) and sources in a faster and easier way. Furthermore, they should obtain the best information for the request, from the most suitable source and in the correct format from all the available information elements.

The main goal of this thesis is to propose a framework for adapting multimodal information retrieval systems' performance based on user behavior (past interactions). This goal is particularized in a system to find **Spanish** multimodal information over heterogeneous sources by defining three intermediary objectives: (i) defining **an Interactive Multimodal Information Retrieval (IMIR) formal model**, (ii) develop-

ing a **basic prototype** for interactive multimodal information retrieval based on the model and (iii) improving the prototype by adding interaction-based **functionality adaptation**.

Several research areas have approached this problem:

- *Federated search*: studies the use of multiple resources (engines) simultaneously. A single query is distributed to different search engines participating in the federation. Federated search then adds the results received from them into a single results' set that is returned to the user. Which retrieval engines are requested and how results are combined is studied in this research area. The lack of semantic knowledge consideration is the main constraint of these works.

- *Aggregated search*: a single retrieval engine is requested but the received results are combined to give a single result to the user. This result must group the relevant and non-redundant information present in every retrieved result. It is limited to return a single result what seems an important drawback. Whenever more than one result is needed, these systems are not appropriate to the scope of this thesis.

- *Multimedia retrieval*: it is focused on retrieving multimedia elements. Most works are based on matching multimedia low-level features, but other use high-level features such as semantic information extracted from the multimedia elements. This area groups the nearest works to our approach.

- *Web search*: it is focused on the study of the retrieval from web engines and their verticals. It is similar to federated search with the difference that every available vertical (engine) is requested with the query without making any distinction. It is interesting to study how do these works select which web engines (or verticals) are requested and how are the results combined. To mention a drawback, there is no semantic relation between the results obtained from different web search engines or verticals.

Chapter 2 has presented some works covering various parts of a possible multimodal model, but every one presents limitations that make them not suitable for our approach. Most similar works to our proposal are:

- The MIMOR model [Womser-Hacker, 1996] stores user relevance feedback to change the long-term optimization of an information retrieval system. The main advantage of MIMOR is that it combines several information retrieval systems. The influence of each system is based upon its previous performance measured by the relevance feedback. The results' fusion is implemented as a weighted linear combination of the individual systems.

- Octopus [Yang et al., 2002] is a multimodal retrieval system that represents multimodal information in a single index with low-level features. They name the index as Multifaceted Knowledge Base (MKB) because it models different levels of knowledge and relevance between media elements. Octopus uses a specific approach defined as Link Analysis based retrieval (LAB) that analyzes links inside MKB and retrieves documents based on them.

- A unified model that aggregates documents, concepts and users is defined in Marchand-Maillet et al. [2011]. It uses propagation strategies and guiding navigation instead of typical searches. Documents are represented in a matrix with documents relations. There are other matrices for concepts and relations between documents and concepts; for users representing the social network; and a last matrix which determines which documents have each user created and rated.

We are interested in retrieval systems requesting multiple engines, so the main drawback of these works is that they use a single retrieval engine (combined index). Requesting a single *RE* makes them not to use handler or results' fusion and limits their impact on our approach. On the contrary, they contain great representations of multimodal elements and semantic knowledge.

The functionality adaptation is the wider and more diversified area. There are plenty of works that adapt systems' functionality to user interactions. The differences between them are the number and type of interactions they consider and the mechanism they use to adapt the functionality. Regarding the interactions, we have introduced them in section 2.7 but the adaptation mechanism is a novelty. Most system are focused on *user modeling* or *personalization*, but our approach tries to improve the IR performance for every user (similar to modeling for a generic user).

It is also latent that there is a need to create systems that adapt to the user. There are many which perform user modeling, but almost none of them do it in a generic

way and without the need to adapt the functionality to specific models. Our proposal adapts to the whole set of users without distinction, what makes the proposal more applicable to new domains.

The improvements carried out in this thesis have tried to solve these problems by (i) defining a formal model, which helps to the definition of a multimodal retrieval system and allows a standardized design of multimodal IR components. The main advantages of following a model is that an unified and standardized framework exists; (ii) implementing a basic multimodal IR prototype based on the previously defined model. This prototype is composed of elements which are easily replaceable by others that have been similarly defined by the model; and (iii) extending the prototype to adapt its functionality to past user interactions. With this extended prototype we covered the need to create a multimodal IR system that adapts its functionality to user behavior.

The results obtained by validating our proposal show that:

- The formal model (see chapter 3) offers an irreplaceable framework for defining different multimodal retrieval systems in order to make them interoperable and interchangeable. This framework can make it easier to define different components for an IMIR system in order to test them easily.

- The implementation of two prototypes based on the formal model in two different scenarios: (a) multimodal retrieval in sports domain and (b) analysis of retrieval information in health social media.

  - The sports-domain prototype is fully functional, as explained in chapter 4. It accepts three query modes, offers five results' visualization techniques and makes use of six retrieval engines. The evaluation only onsiders three retrieval engines (**FTS**, **QAS** and **ObS**). The other engines have not been used because they does not retrieve information from the collections so it is impossible to determine the relevance of the results in the silver standard for these engines. The NDCG (normalized discounted cumulative gain) is obtained for each retrieval engine: **10,1%** (QAS), **80%** (FTS) and **26,8%** (ObS). These results are in the order of the state-of-art works considering multimedia forums like CLEF. When the combination of retrieval engines is

used, the IR performance is increased by a percentage gain of **771,4%** with *QAS*, *7,2%* with *FTS* and **145,5%** with *ObS*.

- The analysis of health social media streams has been completed successfully. It was a big bet to adapt the formal model and to create a functional prototype oriented to a new domain characterized by its specific vocabulary and requirements (see chapter 8). It must be mentioned that the prototype has been implemented with a limited number of characteristics of the formal model, but it constitutes a first approach for us to build a more complex system driven to the health domain.

- Regarding the adaptation of the multimodal IR (chapter 6), we compare the normalized discounted cumulative gain (NDCG) measure obtained with two different approaches: the multimodal system using predefined rules and the same multimodal system once the functionality is adapted by past user interactions. The NDCG has shown an improvement between $-2,92\%$ and $2,81\%$ depending on the approaches used. We have considered three features to classify the approaches (see section 7.1): (i) the classification algorithm; (ii) the query features; and (iii) the scores for computing the orders of retrieval engines. The best result is obtained using probabilities-based classification algorithm (**Probs**), the REs ranking generated with Averaged-Position score (**APS**) and the mode, type, length and entities of the query (**mtle**). Its NDCG value is **81,54%**. By contrast, the worst approach uses K-means classification algorithm (**SKM2**), the mode of the query (**m**) and the REs ranking generated with Interactions-based score (**IbS**) having a NDCG of **76,99%**.

  The first remarkable thing is that the small improvements are conditioned by the good performance obtained when the retrieval engines are used by themselves, that is, it uses only a retrieval engine. This makes the results when combined do not get a big improvement, since each engine has a limited improvement margin. After reviewing all the results, we can conclude that none of them obtains remarkably better results than the other. It is clear that when the query features are not very specific, the results are worse than when they are more specific. This is because the classification algorithms improved the classification task. As regards classification algorithms, there is no significant difference between them. Each

is the best in any of the combinations of scores and query features. Therefore, any of these three classification algorithms could be adopted as a final option to include in the prototype. Something similar happens with measures to determine the score of the engines. Any of them would be worth us to obtain improvements in the IR, being the improvement similar in every case.

9.2 Thesis Impact

The dissemination of the thesis results in the research community is described in this chapter. This dissemination is an indicator of the impact that our developments had. This impact is characterized by two aspects: publications that reflect the impact that this thesis had in the research field and research projects that applied the developments achieved by this thesis.

9.2.1 Publications

The main results from the research work presented throughout this thesis have been disseminated by the publication of the results.

We have three publications at the conference 'Adaptive Multimedia Retrieval' (AMR) in which different works related to this thesis are presented. The first publication [Schneider et al., 2011] was the first contact we had with the retrieval of multimodal information from voice query. It describes the first experiments we conducted with speech recognition systems. In this paper three automatic transcription systems were evaluated.

The second publication [González et al., 2013] developed a methodology for evaluating automatic transcription systems. This methodology was necessary because when we tried to evaluate automatic speech recognizers at our disposal for inclusion in the development of this thesis, we realized that no formal way to evaluate an ASR was defined. Therefore, the methodology was defined in order to facilitate the standardization of the assessment process. This methodology was used in this thesis to determine which of the available ASR was to be used and included in the basic prototype (see section 4.2.5).

The third publication [Schneider et al., 2014] used a system for correcting named entities in voice queries. This work was done because we observed that the results of

information retrieval based on voice queries was very bad. we realized that in most cases it was motivated by errors in the transcription of the query, not because the information retrieval system was malfunctioning. Therefore, correction of these failures seemed necessary. The results showed that a domain-specific system could provide improvements in the percentage of correct transcribed words around 25% (see section 4.2.4).

Another conference we presented works to is Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural - SEPLN (Congress of the Spanish Society for Natural Language Processing). The first article we submitted [del Valle et al., 2008] presented a multimodal system that was developed under the scope of a collaboration with SOPAT project (CIT-410000-2007-12). It was based on the use of a speech recognition system and a speech synthesis for guiding people (visually impaired) through a hotel. This system served as an entry point to the retrieval of multimodal information, giving rise to many questions that led us to make this thesis. It was also the starting point for our interest in voice technology, both recognizer and synthesizer.

In 2013 we presented another paper to SEPLN. This paper [Schneider et al., 2013] explains the application of ontologies in financial domains to a query expansion process. Its final goal is to improve financial information retrieval effectiveness. The system is composed of an ontology and a Lucene index that stores and retrieves natural language concepts. This work contributes to the integration of ontologies as retrieval engines in the thesis development.

As far as evaluation forums is concerned, evaluation forums are conference parts that are focused on presenting benchmarks to evaluate comparatively several systems. In this sense, we have actively participated in several evaluation forums during the thesis development. Mainly, our participation is focus on the question answering track of CLEF (Cross-Language Evaluation Forum). We participated two years in the QA track with two different approaches: first we tried to use passages of documents in order to analyze if they contained enough information to perform retrieval based on them [Vicente-Díez et al., 2009]; the second year we analyzed temporal implications of documents in order to improve retrieval Vicente-Díez et al. [2010a]. The two-year results obtained fairly low, but helped us to deepen knowledge we had of retrieval systems. In addition to these systems, we also participate in TempEval task of Semeval2010 by analyzing temporal expressions in Spanish texts [Vicente-Díez et al., 2010b]. This

collaboration perfected my knowledge of NLP as well as allowing us to improve the above mentioned QA system.

As far as journals is concerned, we presented an article to the BMC Medical Informatics and Decision Making journal [Bedmar et al., 2015], where we present a system for detecting drug effects (which include both adverse drug reactions as well as drug indications) from user messages collected from a Spanish health social network. Texts were processed using MeaningCloud[130], a multilingual text analysis engine, to identify drugs and effects. We then applied a distant-supervision method using the database on a collection of 84,000 messages in order to extract the relations between drugs and their effects. To classify the relation instances, we used a kernel method based only on shallow linguistic information of the sentences. The relation of this work with our thesis is mostly oriented to the application of our results, first in a new domain such as biomedicine, and secondly, to retrieve information from social networks.

9.2.2 Research and Development (R&D) projects

This thesis has been carried out under the scope and influence of the next projects:

- **Trendminer**[131] (FP7-ICT 287863) is a research project that studies real-time social media streams. Healthcare is one of the domains it covers, building a system to detect relations between drugs, adverse effects and diseases, as well as extracting usage statistics from media streams. It uses Twitter and health-related forums (Saluspot[132] and Forumclinic[133]). This project has supported the introduction of social media streams as information sources (through retrieval engines). Social media streams are real-time information generators, indeed, the management of the information generated by them should be processed in real-time in order to make this information available as soon as possible. The possibility of applying multimodal retrieval in a new domain (healthcare) is the second contribution of the project.

[130] http://www.meaningcloud.com/es/
[131] http://www.trendminer-project.eu/
[132] https://www.saluspot.com/
[133] http://www.forumclinic.org/

- **Buscamedia**[134] (CEN-20091026) is focused on multimedia information retrieval. Its main goal is the definition and implementation of a multimedia retrieval system that can accept multimedia queries in order to obtain multimedia elements as response. An ontology is used to offer semantic knowledge. This project has served as an inspiration for the development of the thesis. Although the topic of the thesis was defined, the broad field of multimodal information retrieval forced us to limit the coverage we wanted to do, therefore the project requirements have been adopted for the definition of the objectives. Requirements is not the only thing we took from this project, but in the evaluation we have used the collection of documents that has been developed within the scope of this project.

- **Bravo (Búsqueda de Respuestas Avanzada Multimodal y Multilingüe)** (TIN2007-67407-C03-01) was dedicated to research in technologies to improve the search for answers to both input text or voice. It served as a beginning for our interest in voice queries. In this project we performed voice recognition analysis as input to a question & answering (QA) system. The information retrieval system developed in this project was our first contact with multimodal information retrieval, because it allowed the use of two modes in the query. Therefore, it forced us to use a sequential retrieval system where first the transcription of the query was obtained and then used like a text.

- **MAVIR**[135] **(S-0505/TIC-0267) and MAVIR2 (S-2009/TIC-1542)**: are not projects but consortia that had the purpose of sharing the knowledge generated by the projects of their partners and to foster collaboration in R&D in the Community of Madrid.

9.3 Future lines

In this chapter we will focus on two different future lines (shown in figure 9.1): (i) improvements or extensions that can be performed on the developments carried out during this thesis; and (ii) the application of these developments to other domains (or research areas).

[134] http://www.cenitbuscamedia.es/
[135] http://www.mavir.net/

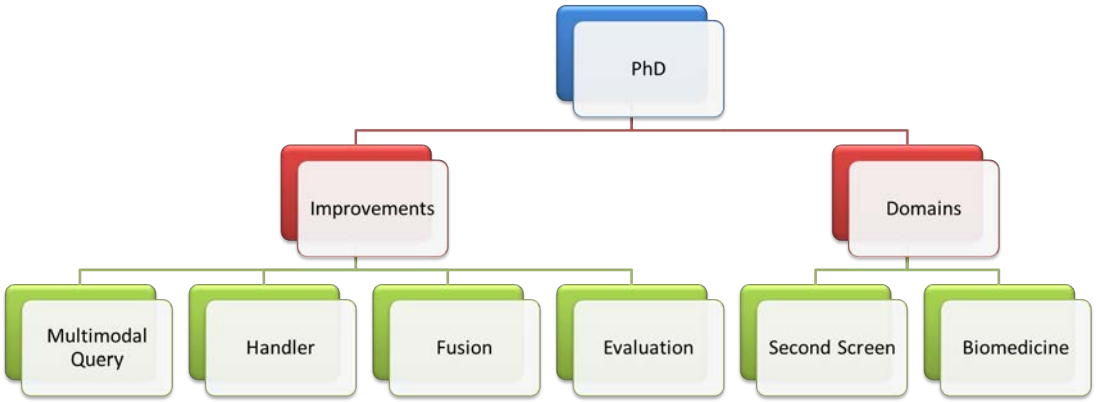


Figure 9.1: Future lines organization

9.3.1 Improvements over thesis developments

Regarding the formal model, there are a couple of points that have not been fully integrated and can be studied and addressed in future model improvements or expansions. The main point that has been simplified is the source selection strategies (handler) and the fusion results' techniques. These modules have limited their definition to rule-based systems, although they can be of any type, so this definition should be generalized. This led us to postpone the general definition and propose it as a possible future line to continue with.

The results' fusion has the limitation having been defined as a matrix product. Then, the size of each results' set is forced to be equal (see equation 3.20). This should be modified to avoid this limitation.

Looking at the prototype, every implemented component can be improved or modified, so many research branches can be defined here. Specifically, more deep results' merging or fusion techniques can be analyzed.

The future lines regarding user behavior adaptation are mainly defined by the results. The results show that adjusting a multimodal retrieval system using historical behavior of users improves IR results, but our approach presents limitations:

- We have only researched the use of two user interactions: *results browsing* and *relevance judgments*. The number of historical behavior characteristics and the

processing done on them can be studied.

- Another limitation of the algorithm is that only five similarity measurements have been defined and used. Generating new similarity measures to generate rules is also a possible future research option. (New engine scores or new ways of generating rules in the silverstandard; new AI algorithms such as Bayesian networks or proper algorithm; or even trying more query characteristics to perform the classification).

- A more exhaustive evaluation with end users would be interesting. The work of Kelly [2007] can be used as a initial point to define a task-oriented evaluation that can engage a bigger number of users that make more interactions. This can help to increase the number of interactions used to train the adaptation models.

Although **multimodal query** is considered inside the formal model, the thesis has not studied and developed every characteristic of this type of queries. The prototype has been fixed to use three types of queries and a whole study about a wider number of modes and combinations could be interesting as a new study. Some works have studied it (such as Querium system [Golovchinsky and Diriye, 2011]) but we thought that multimodal query will be an important part in future IR systems. Thus, much more research can be done in this line.

9.3.2 New areas of application

The application of the learned lessons (model, prototype, adaptation) in new domains seems to be the most commercial future line. There are two up-to-date domains for which the application of the model, the prototype and the adaptation perfectly fits:

- A **second screen** is *'a second electronic device used by television viewers to connect to a program they're watching. A second screen is often a smartphone or tablet, where a special complementary app may allow the viewer to interact with a television program in a different way - the tablet or smartphone becomes a TV companion device. The second screen phenomenon represents an attempt to make TV more interactive for viewers, and help promote social buzz around specific programs'*[136]. Second screen is becoming popular for users watching TV.

[136]Taken from http://www.techopedia.com/definition/29212/second-screen

The Digital Consumer Report 2014 Nielsen[137] claims that 66% tablet and 49% smartphone owners surf the web while watching TV. Between the most common usage are: shopping, checking sports scores, email/text friends about the program and look up information about actors, plotlines, or athletes. The last usage can be performed using a multimodal retrieval system. The different queries that users can handle are presented in various modes: texts, images, audios, videos, graphics, etc. This domain presents the problem that real time is essential, because depending on what you are watching (first screen), the topic can change very quickly (like in broadcast news). Visualization techniques are especially important because the information must be easily visible together with the content of the first screen.

- **Health social media streams analysis** is an up-to-date domain that currently is focusing plenty of research works. It is an interesting domain when talking about multimedia retrieval because it handles many different information modes: clinic reports (text), X-ray (images) or ultrasound (video). Time constrains are very important. The faster a doctor or a patient gets the information, the more adequate the treatment could be. The architecture displayed at figure 4.1 is directly applicable by making some easy modifications:

  1. Changing external REs or the collections that are currently being used.
  2. Adapting the rules of the handler to the new query types (if they are different in the domain) and the new REs.
  3. Studying if the fusion strategy is usable or defining a new fusion strategy.
  4. Analyzing if some new interactions must be considered.

- Linking Open Data [Florian and Martin, 2012] (LOD): Linked Data is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods. More specifically, Wikipedia defines Linked Data as "a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF."

[137] http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2014%20Reports/the-digital-consumer-report-feb-2014.pdf

The current state of the Linking Open Data cloud can be visualized at `http://lod-cloud.net/`.

Appendix A

Annex 1

A.1 Final User Survey

The questions of the final survey must be answered with a numeric value from 0 (being the worst) to 5 (being the best). The questions are explained next:

1. To what extent is the system easy to use?

   This question intended to determine to what extentd the system was easy to use from the users point of view.

2. If you have performed Textual Queries, In what degree have you been able to formulate your queries?

3. If you have performed Textual Queries accompanied by an Image, In what degree have you been able to formulate your queries?

4. If you have performed Voice Queries, In what degree have you been able to formulate your queries?

   The previous three questions asked for the perception of the user about the query boxes, i.e., if it was easy to generate queries, specially the not common queries (multimodal query combining text and image and audio query).

5. Which is your degree of satisfaction with the way the system shows the results of a query?

   The user is supposed to score the combined list of results that is given initially (as the first visualization) after each search.

6. Which is your degree of satisfaction with the way the system shows Groups of terms as a result of a query?

   The user should score the cloud of terms (both concepts and answers).

7. Which is your degree of satisfaction with the way the system shows Groups of concepts as a result of a query?

   The user is asked to value the semantic grouping visualization.

8. In which degree did the system solve your information needs?

   Users must give their overall opinion about the system, valuing from the graphical user interface to the retrieval performance.

9. Relating to the search engine you have evaluated, Which are the most important characteristics?

   This questions tried to value the initial level of knowledge that users had about information retrieval systems.

10. What elements do you miss in the system?

    This was a question that has a non-numeric answer, i.e., users must write their opinion about the lacks of the system.

11. Have you had any problems with voice search? If so, could you describe it?

    This was also a question that has a non-numeric answer, i.e., users must write their problems using the audio transcription engine.

Appendix B

Annex 2: Audio Transcription Details

B.1 XML Dictionary Structure

The implemented dictionary is composed by a set of named entities together with its associated information (in XML format). It structure can be seen in table B.1.

```
<dictionary>
<properties>
<totalentities>2000</totalentities>
<totalFP>1900</totalFP>
<totalFS>42</totalFS>
<totalFT>42</totalFT>
<searchedentities>2345</searchedentities>
<searchedFP>2200</searchedFP>
<searchedFS>85</searchedFS>
<searchedFT>60</searchedFT>
</properties>
<entities
<entity>
<text>Lionel Messi</text>
<type>FootballPlayer</type>
<popularity>0.9</popularity>
<historic>4</historic>
</entity>
...
</entities>
</dictionary>
```

Table B.1: Example of Named Entity stored in Dictionary

References

Agichtein, E., Brill, E., and Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 19–26, New York, NY, USA. ACM. 59

Ahn, J., Wongsuphasawat, K., and Brusilovsky, P. (2011). Analyzing user behavior patterns in adaptive exploratory search systems with lifeflow. In *The Fifth Workshop on Human-Computer Interaction and Information Retrieval*. 23, 27, 32, 38, 57

Ahn, J.-w., Brusilovsky, P., He, D., Grady, J., and Li, Q. (2008). Personalized web exploration with task models. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 1–10, New York, NY, USA. ACM. 38, 57

Ahn, J.-w. and Btusilovsky, P. (2009). Adaptive visualization of search results: Bringing user models to visual analytics. *Information Visualization*, 8(3):167–179. 38, 57

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65. 81

Arampatzis, A., Zagoris, K., and Chatzichristofis, S. A. (2011). Fusion vs. two-stage for multimodal retrieval. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 759–762, Berlin, Heidelberg. Springer-Verlag. 9, 24, 27, 44, 47, 50

Arguello, J., Diaz, F., and Callan, J. (2011). Learning to aggregate vertical results into web search results. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 201–210, New York, NY, USA. ACM. 10

Arguello, J., Wu, W.-C., Kelly, D., and Edwards, A. (2012). Task complexity, vertical display and user interaction in aggregated search. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 435–444, New York, NY, USA. ACM. vii, 30, 32, 39, 42, 48, 56, 61, 66, 129

Balog, K. (2013). Collection and document langauge models for resrouce selection. In *Proceedings of the Twenty-second Text Retrieval Conference (TREC)*. To appear. 62, 68, 161

Balog, K., Neumayer, R., and Nørvåg, K. (2012). Collection ranking and selection for federated entity search. In *Proceedings of the 19th international conference on String Processing and Information Retrieval*, SPIRE'12, pages 73–85, Berlin, Heidelberg. Springer-Verlag. 24, 27, 43, 45, 46, 48, 50

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359. 111

Beckers, T. and Fuhr, N. (2010). User-oriented and eye-tracking-based evaluation of an interactive search system. In *4th Workshop on Human-Computer Interaction and Information Retrieval (HCIR 2010) @ IIiX 2010*. 32, 38, 60, 66

Bedmar, I. S., Martínez, P., Arenaz, R. R., and Schneider, J. M. (2015). Exploring spanish health social media for detecting drug effects. *BMC Medical Informatics and Decision Making*, 15. 183, 216

Bellogin, A., Gebremeskel, G. G., He, J., Said, A., Samar, T., de Vries, A. P., Lin, J., and Vuurens, J. B. P. (2013). Cwi and tu delft notebook trec 2013: Contextual suggestion, federated web search, kba, and web tracks. In *Proceedings of the Twenty-second Text Retrieval Conference (TREC)*. To appear. 44

Ben-Gal, I. (2007). *Bayesian Networks*. John Wiley and Sons. 61

Bernsen, N. O., Dybkjær, L., and Minker, W. (2007). *Spoken Dialogue Systems Evaluation*, pages 187–219. Text, Speech and Language Technology. Springer, Dordrecht (The Netherlands), Evaluation of Text and Speech Systems edition. 116

Bessai-Mechmache, F. Z. and Alimazighi, Z. (2012). Possibilistic model for aggregated search in xml documents. *Int. J. Intell. Inf. Database Syst.*, 6(4):381–404. 23, 27, 47, 66

Bond, C. and Raehl, C. (2006). Adverse drug reactions in united states hospitals. *Pharmacotherapy*, 5(26):601–608. 184

Bota, H., Zhou, K., Jose, J. M., and Lalmas, M. (2014). Composite retrieval of heterogeneous web search. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 119–130, New York, NY, USA. ACM. 49

Boyce, R., Ryan, P., Norén, G., Schuemie, M., Reich, C., Duke, J., Tatonetti, N., Trifir, G., Harpaz, R., Overhage, J., Hartzema, A., Khayter, M., Voss, E., Lambert, C., Huser, V., and Dumontier, M. (2014). Bridging islands of information to establish an integrated knowledge base of drugs and health outcomes of interest. *Drug Safety*, 37(8):557–567. 191

Bracamonte, T. (2013). Multimedia information retrieval on the social web. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 349–354, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee. 53

Breiman, L. (2001). Random forests. *Mach. Learn.*, 45(1):5–32. 58

Buccio, E. D., Masiero, I., and Melucci, M. (2013). University of padua at trec 2013: Federated web search track. In *Proceedings of the Twenty-second Text Retrieval Conference (TREC)*. To appear. 44

Buccio, E. D., Melucci, M., and Song, D. (2010). Exploring combinations of sources for interaction features for document re-ranking. In *4th Workshop on Human-Computer Interaction and Information Retrieval (HCIR2010)*. pp.63-66. 23, 59, 60

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 89–96, New York, NY, USA. ACM. 59

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167. 110

Cabot, J. and Clarisó, R. (2014). Evaluating the quality of software models using light-weight formal methods. *ERCIM News*, 42(99):29–30. 92

Caicedo, J. C. (2009). Multimodal information spaces for content-based image retrieval. In *Proceedings of the Third BCS-IRSG conference on Future Directions in Information Access*, FDIA'09, pages 110–116, Swinton, UK, UK. British Computer Society. 24, 32

Callan, J., Croft, W., and Harding, S. (1992). The inquery retrieval system. In Tjoa, A. M. and Ramos, I., editors, *Database and Expert Systems Applications*, pages 78–83. Springer Vienna. 23

Castells, P., Fernandez, M., and Vallet, D. (2007). An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Trans. on Knowl. and Data Eng.*, 19(2):261–272. 52

Cavanagh, J. M. A. (1976). *The Library Quarterly: Information, Community, Policy*, 46(1):pp. 79–81. 37, 81

Chernov, S., Kohlschütter, C., and Nejdl, W. (2006). A plugin architecture enabling federated search for digital libraries. In *Proceedings of the 9th international conference on Asian Digital Libraries: achievements, Challenges and Opportunities*, ICADL'06, pages 202–211, Berlin, Heidelberg. Springer-Verlag. 10, 23, 42, 46, 47

Cintra, M. E., Monard, M. C., and Camargo, H. A. (2013). A fuzzy decision tree algorithm based on C4.5. *Mathware & Soft Computing Magazine. The Magazine of the European Society for Fuzzy Logic and Technology*, 20:56–62. 153

Cohen, W. W., Ravikumar, P., and Fienberg, S. E. (2003). A comparison of string distance metrics for name-matching tasks. pages 73–78. 121

Cole, M. J., Gwizdka, J., Belkin, N. J., and Liu, C. (2011). User Domain Knowledge and Eye Movement Patterns During Search. In *The Fifth Workshop on Human-Computer Interaction and Information Retrieval*. 60

Daras, P., Axenopoulos, A., Darlagiannis, V., Tzovaras, D., Bourdon, X. L., Joyeux, L., Verroust-Blondet, A., Croce, V., Steiner, T., Massari, A., Camurri, A., Morin, S., Mezaour, A.-D., Sutton, L., and Spiller, S. (2011). Introducing a unified framework for content object description. *IJMIS*, 2(3/4):351–375. 30, 34

de Vries, A. (1998). Mirror: Multimedia query processing in extensible databases. xi, 9, 30, 31, 35, 36, 57, 67

del Valle, D., Rivero, J., Conde, D., Olaziregi, G., Moreno, J., Calle, F. J., and Cuadra, D. (2008). Plataforma de interacción natural para el acompañamiento virtual. *Procesamiento del Lenguaje Natural*, 41. 215

Demeester, T., Trieschnigg, D., Nguyen, D., and Hiemstra, D. (2013). Overview of the trec 2013 federated web search track. 16, 64, 65

Demner-Fushman, D., Antani, S., Simpson, M. S., and Thoma, G. R. (2012). Design and development of a multimodal biomedical information retrieval system. *JCSE*, 6(2):168–177. 9, 24, 27, 34, 40, 42, 46, 49, 50, 68

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. (2001). Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 613–622, New York, NY, USA. ACM. 45

Florian, B. and Martin, K. (2012). *Linked Open Data: The Essentials - A Quick Start Guide for Decision Makers.* edition mono/monochrom, Vienna, Austria. 220

Frank van Harmelen and Peter F. Patel-Schneider and Ian Horrocks (2001). Reference description of the daml+oil (march 2001) ontology markup language. http://www.daml.org/2001/03/reference.html. 30, 34

Freifeld, C., Brownstein, J., Menone, C., Bao, W., Filice, R., Kass-Hout, T., and Dasgupta, N. (2014). Digital drug safety surveillance: Monitoring pharmaceutical products in twitter. *Drug Safety*, 37(5):343–350. 186

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969. 58

Galiano, M. C. D. (2011). *Recuperación de información multimodal basada en integración de conocimiento.* PhD thesis. 84

Galiano, M. C. D., Valdivia, M. T. M., Rez, r. M., and Lpez, L. A. U. (2007). Mejora de los sistemas multimodales mediante el uso de ganancia de informacin. *Procesamiento del lenguaje natural*, 38:119–130. 9

García, R. and Celma, O. (2005). Semantic integration and retrieval of multimedia metadata. In Handschuh, S., Declerck, T., and Koivunen, M., editors, *Proceedings of the ISWC 2005 Workshop on Knowledge Markup and Semantic Annotation (Semannot'2005)*, volume 185, pages 69–80. CEUR Workshop Proceedings. 54

Gerl, M., Rautek, P., Isenberg, T., and Groeller, M. E. (2012). Semantics by analogy for illustrative volume visualization. *Computers and Graphics*, 36(3):201–213. vii, 132

Gil, J. (2007). *Transcripción fonética: Representación escrita de los sonidos que pronunciamos*, page 547. Text, Speech and Language Technology. Arco/Libros, Fonética para profesores de español: De la teoría a la práctica edition. 121

Golovchinsky, G. and Diriye, A. (2011). Session-based search with querium. In *HCIR 2011.* 23, 56, 219

González, M., Moreno Schneider, J., Martínez, J. L., and Martínez, P. (2013). An illustrated methodology for evaluating asr systems. In *Proceedings of the 9th International Conference on Adaptive Multimedia Retrieval: Large-scale Multimedia Retrieval and Evaluation*, AMR'11, pages 33–42, Berlin, Heidelberg. Springer-Verlag. 115, 214

Görg, C., Kihm, J., Choo, J., Liu, Z., Muthiah, S., Park, H., and Stasko, J. (2010). Combining Computational Analyses and Interactive Visualization to Enhance Information Retrieval. In *2010 Workshop on Human-Computer Interaction and Information Retrieval, New Brunswick, NJ, August 2010.* 32, 38, 66

Guan, F., Xue, Y., Yu, X., Liu, Y., and Cheng, X. (2013). Ictnet at federated web search track 2013. In *Proceedings of the Twenty-second Text Retrieval Conference (TREC).* To appear. 49, 50

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, NY, USA. 121

Gutiérrez, P. A., Hervás-Martínez, C., and Lozano, M. (2010). Designing Multilayer Perceptrons using a Guided Saw-tooth Evolutionary Programming Algorithm. *Soft Computing*, 14(6):599–613. 153

Halvey, M. J. and Keane, M. T. (2007). An assessment of tag presentation techniques. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 1313–1314, New York, NY, USA. ACM. vii, 132

Hamers, L., Hemeryck, Y., Herweyers, G., Janssen, M., Keters, H., Rousseau, R., and Vanhoutte, A. (1989). Similarity measures in scientometric research: The jaccard index versus salton's cosine formula. *Inf. Process. Manage.*, 25(3):315–318. 45

Hannan, J. (1957). Approximation to Bayes risk in repeated plays. *Contributions to the Theory of Games*, 3:97–139. 56

Harpaz, R., DuMouchel, W., Shah, N. H., Madigan, D., Ryan, P., and Friedman, C. (2012). Novel data-mining methodologies for adverse drug event discovery and analysis. *Clin. Pharmacol. Ther.*, 91(6). 186

Hauptmann, A. G., Jin, R., and Ng, T. D. (2002). Multi-modal information retrieval from broadcast video using ocr and speech recognition. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, JCDL '02, pages 160–161, New York, NY, USA. ACM. 33, 39, 43, 45, 68

Hawking, D. and Craswell, N. (2001). Overview of the TREC-2001 Web Track. In *Proceedings of TREC-2001*. `http://david-hawking.net/pubs/hawking_trec01wt.pdf`. 23

Herxheimer, A., Crombag, M., and Alves, T. (2010). Direct patient reporting of adverse drug reactions. a twelve-country survey & literature review. *Health Action International (HAI)*, 01(01):41–49. 185

Hodges, S., Williams, L., Berry, E., Izadi, S., Srinivasan, J., Butler, A., Smyth, G., Kapur, N., and Wood, K. (2006). Sensecam: A retrospective memory aid. In *Proceedings of the 8th International Conference on Ubiquitous Computing*, UbiComp'06, pages 177–193, Berlin, Heidelberg. Springer-Verlag. 24

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, 42(1-2):177–196. 38

Hoi, S. C. and Wu, P. (2011). Sire: a social image retrieval engine. In *Proceedings of the 19th ACM international conference on Multimedia*, MM '11, pages 817–818, New York, NY, USA. ACM. 39, 43, 59, 67

Hoi, S. C. H., Jin, R., Zhu, J., and Lyu, M. R. (2009). Semisupervised svm batch mode active learning with applications to image retrieval. *ACM Trans. Inf. Syst.*, 27(3):16:1–16:29. 59

Hong, D. and Si, L. (2012). Mixture model with multiple centralized retrieval algorithms for result merging in federated search. In Hersh, W. R., Callan, J., Maarek, Y., and Sanderson, M., editors, *SIGIR*, pages 821–830. ACM. 10, 23, 32, 38, 42, 47, 67

Hu, X., Kando, N., and Yuan, X. (2011). User evaluation of an interactive music information retrieval system. In *In Proceedings of HCIR2011 Workshop, Mountain View, CA, USA*. 25, 32, 40, 43, 66

Huang, J. (2011). On the value of page-level interactions in web search. In *In Proceedings of HCIR2011 Workshop, Mountain View, CA, USA*. 60

Ide, N. C., Loane, R. F., and Demner-Fushman, D. (2007). Essie: a concept-based search engine for structured biomedical text. *J Am Med Inform Assoc*, 14(3):253–263. 40

iSoco (2013). Buscamedia m3 ontology network and service. 101

Itti, L. and Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40:1489–1506. 111

Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual*

International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05, pages 154–161, New York, NY, USA. ACM. 60, 62

Jolliffe, I. (1986). *Principal Component Analysis*. Springer Verlag. 59

Jones, K. S., Walker, S., and Robertson, S. E. (2000). A probabilistic model of information retrieval: Development and comparative experiments. *Inf. Process. Manage.*, 36(6):779–808. 37, 49, 81

Jou, B., Li, H., Ellis, J. G., Morozoff-Abegauz, D., and Chang, S.-F. (2013). Structured exploration of who, what, when, and where in heterogeneous multimedia news sources. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 357–360, New York, NY, USA. ACM. 25, 27, 43

Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *J. Mach. Learn. Res.*, 8:613–636. 61

Kalpathy-Cramer, J., Müller, H., Bedrick, S., Eggel, I., García Seco de Herrera, A., and Tsikrika, T. (2011). The CLEF 2011 medical image retrieval and classification tasks. In *Working Notes of CLEF 2011 (Cross Language Evaluation Forum)*. 24

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):881–892. 153

Kaplan, A. M. and Haenlein, M. (2010). Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 53(1):59 – 68. 183

Kelly, D. (2007). Methods for Evaluating Interactive Information Retrieval Systems with Users. *Foundations and Trends in Information Retrieval*, 3(1 2):1–224. 136, 219

Kelly, D. and Sugimoto, C. R. (2013). A systematic review of interactive information retrieval evaluation studies, 19672006. *Journal of the American Society for Information Science and Technology*, 64(4):745–770. 141

Kludas, J., Bruno, E., and Marchand-Maillet, S. (2008). Adaptive multimedial retrieval: Retrieval, user, and semantics. chapter Information Fusion in Multimedia Information Retrieval, pages 147–159. Springer-Verlag, Berlin, Heidelberg. 9

Kludas, J. and Marchand-Maillet, S. (2011). Effective multimodal information fusion by structure learning. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8. 24

Kong, W., Aktolga, E., and Allan, J. (2013). Improving passage ranking with user behavior information. In He, Q., Iyengar, A., Nejdl, W., Pei, J., and Rastogi, R., editors, *CIKM*, pages 1999–2008. ACM. 58, 62

Kules, B., Capra, R., Banta, M., and Sierra, T. (2009). What do exploratory searchers look at in a faceted search interface? In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '09, pages 313–322, New York, NY, USA. ACM. 60, 61

Lana-Serrano, S., Villena-Román, J., and Cristóbal, J. C. G. (2011). Daedalus at imageclef medical retrieval 2011: Textual, visual and multimodal experiments. In Petras, V., Forner, P., and Clough, P. D., editors, *CLEF (Notebook Papers/Labs/Workshop)*. 24, 27, 40, 43, 45

Liu, C., Cole, M., Belkin, N., Gwizdka, J., and Zhang, X. (2011). Exploring the Effect of Task Difficulty and Domain Knowledge on Dwell times. In *The Fifth Workshop on Human-Computer Interaction and Information Retrieval*. 39, 60

LivingSpanish (2011). LivingSpanish: Correspondencia de fonemas y grafías en español. `http://www.livingspanish.com/correspondencia-fonetica-grafia.htm`. Accessed: 2011. 121

Lopez, V., Fernández, M., Motta, E., and Stieler, N. (2012). Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic Web*, 3(3):249–265. 53

Lux, M. (2011). Content based image retrieval with lire. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 735–738, New York, NY, USA. ACM. 40, 41

Malla, R., Choudhury, M., and Bali, K. (2011). Web searching with visual clues: A user study. In *In Proceedings of HCIR2011 Workshop, Mountain View, CA, USA*. 30, 32, 34, 39, 42, 61, 66, 84

Mandl, T. and Womser-Hacker, C. (2003). A content independent model for context adaptation and individualization in information retrieval. 57, 58, 62

Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., and Yamada, A. (2001). Color and texture descriptors. *IEEE Trans. Cir. and Sys. for Video Technol.*, 11(6):703–715. 110

Marchand-Maillet, S., Morrison, D., Szekely, E., Kludas, J., Vonwyl, M., and Bruno, E. (2011). Mining networked media collections. In Detyniecki, M., Garca-Serrano, A., and Nrnberger, A., editors, *Adaptive Multimedia Retrieval. Understanding Media and Adapting to the User*, volume 6535 of *Lecture Notes in Computer Science*, pages 1–11. Springer Berlin Heidelberg. 25, 28, 30, 31, 34, 36, 41, 49, 50, 68, 211

Martínez, P., Fernández, J. L. M., Bedmar, I. S., Schneider, J. M., Luna, A., and Arenaz, R. R. (2015). Turning user generated health-related content into actionable knowledge through text analytics services. *Computers in Industry*, 15. 183

Martínez-Fernández, J. L., Martínez, P., Ogrodniczuk, M., and Miháltz, M. (2014). Newly generated domain-specific language data and tools. Technical Report D10.1-287863, DFKI. 183, 198

Martínez-González, A., Pablo-Sánchez, C., Polo-Bayo, C., Vicente-Díez, M., Martínez-Fernández, P., and Martínez-Fernández, J. (2009). The miracle team at the clef 2008 multilingual question answering track. In Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G., Kurimo, M., Mandl, T., Peñas, A., and Petras, V., editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of *Lecture Notes in Computer Science*, pages 409–420. Springer Berlin Heidelberg. 140

McCandless, M., Hatcher, E., and Gospodnetic, O. (2010). *Lucene in Action, Second Edition: Covers Apache Lucene 3.0.* Manning Publications Co., Greenwich, CT, USA. 40, 106

McClellan, M. (2007). Drug safety reform at the fda-pendulum swing or systematic improvement? *N Engl J Med*, 17(356):1700–1702. 185

Medina-Ramírez, R. C. (2007). Semantic information retrieval: a return on experience. *Engineering Letters*, 15(2):234 – 239. 52, 82

Mianowska, B. and Nguyen, N. (2011). A method for user profile adaptation in document retrieval. In Nguyen, N., Kim, C.-G., and Janiak, A., editors, *Intelligent Information and Database Systems*, volume 6592 of *Lecture Notes in Computer Science*, pages 181–192. Springer Berlin Heidelberg. 55

Miguel, J. and Magalhes, C. (2008). *Statistical Models for Semantic-Multimedia Information Retrieval*. PhD thesis, University of London. Imperial College of Science, Technology and Medicine. Department of Computing. 10

Montague, M. and Aslam, J. A. (2002). Condorcet fusion for improved retrieval. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, pages 538–548, New York, NY, USA. ACM. 49

Mourao, A. and Magalhaes, F. M. J. (2013). Novasearch at trec 2013 federated web search track: Experiments with rank fusion. In *Proceedings of the Twenty-second Text Retrieval Conference (TREC)*. To appear. 49

Nottelmann, H. and Fuhr, N. (2003). The mind architecture for heterogeneous multimedia federated digital libraries. In Callan, J. P., Crestani, F., and Sanderson, M., editors, *Distributed Multimedia Information Retrieval*, volume 2924 of *Lecture Notes in Computer Science*, pages 112–125. Springer. v, 9, 10, 30, 31, 34, 35, 47, 48, 67

Olvera Lobo, M. D. (1999). Evaluación de sistemas de recuperación de información : aproximaciones y nuevas tendencias. *El Profesional de la Información*, 8(11). 135

Pablo-Sánchez, C., Martínez-Fernández, J., González-Ledesma, A., Samy, D., Martínez, P., Moreno-Sandoval, A., and Al-Jumaily, H. (2008). Combining wikipedia and newswire texts for question answering in spanish. In Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D., Peñas, A., Petras, V., and Santos, D., editors, *Advances in Multilingual and Multimodal Information Retrieval*, volume 5152 of *Lecture Notes in Computer Science*, pages 352–355. Springer Berlin Heidelberg. 140

Pal, D. and Mitra, M. (2013). Isi at the trec 2013 federated task. In *Proceedings of the Twenty-second Text Retrieval Conference (TREC)*. To appear. 44, 49, 50, 161

Paris, C., Wan, S., and Thomas, P. (2010). Focused and aggregated search: a perspective from natural language generation. *Inf. Retr.*, 13(5):434–459. 10

Pérez-Iglesias, J., Pérez-Agüera, J. R., Fresno, V., and Feinstein, Y. Z. (2009). Integrating the Probabilistic Models BM25/BM25F into Lucene. *CoRR*, abs/0911.5046. 106

Pino, C. and Di Salvo, R. (2011). A survey of semantic multimedia retrieval systems. In *Proceedings of the 13th WSEAS International Conference on Mathematical and Computational Methods in Science and Engineering*, MACMESE'11, pages 353–358, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS). 51, 53

Platt, J. C. (1999). Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. MIT Press, Cambridge, MA, USA. 40

Porter, M. F. (2001). Snowball: A language for stemming algorithms. Published online. Accessed 11.03.2008, 15.00h. 106

Project, A. C. R. and Cleverdon, C. (1962). *Report on the Testing and Analysis of an Investigation Into Comparative Efficiency of Indexing Systems*. College of Aeronautics. 166

Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.*, 1(1):81–106. 153

Rekha, C., Usharani, J., and Iyakutti, K. (2011). Article: Improving the information retrieval system through effective evaluation of web page in client side analysis. *International Journal of Computer Applications*, 15(6):35–39. Published by Foundation of Computer Science. 56, 62

Renaud, G. and Azzopardi, L. (2012). Scamp: a tool for conducting interactive information retrieval experiments. In *Proceedings of the 4th Information Interaction in Context Symposium*, IIIX '12, pages 286–289, New York, NY, USA. ACM. 30, 42, 46, 61, 67, 68, 154

Robertson, S. E., Walker, S., Beaulieu, M. H., Gull, A., and Lau, M. (1992). Okapi at trec. In *Text REtrieval Conference*, pages 21–30. 39

Romberg, S., Lienhart, R., and Hörster, E. (2012). Multimodal image retrieval - fusing modalities with multilayer multimodal plsa. *IJMIR*, 1(1):31–44. 24, 27, 38, 42, 46, 47, 48, 50, 67

Rotella, F., Ferilli, S., and Leuzzi, F. (2013). An approach to automated learning of conceptual graphs from text. pages 341–350. 77

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620. 37, 81

Santos Jr, E. and Nguyen, H. (2009). Modeling users for adaptive information retrieval by capturing user intent. *Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling. IGI Global*, pages 88–118. 60, 62

Sato, T., Kanade, T., Hughes, E. K., and Smith, M. A. (1998). Video ocr for digital news archive. In *CAIVD*, pages 52–60. 39

Schneider, J. M., Declerck, T., Fernández, J. L. M., and Martínez, P. (2013). Prueba de concepto de expansión de consultas basada en ontologías de dominio financiero. *Procesamiento del Lenguaje Natural*, 51:109–116. 215

Schneider, J. M., Fernández, J. L. M., and Martínez, P. (2014). A proof-of-concept for orthographic named entity correction in spanish voice queries. In Nrnberger, A., Stober, S., Larsen, B., and Detyniecki, M., editors, *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, Lecture Notes in Computer Science, pages 181–190. Springer International Publishing. vii, 117, 118, 119, 214

Schneider, J. M., Salazar, M. G., Martínez, P., and Fernández, J. L. M. (2009). Some experiments in evaluating asr systems applied to multimedia retrieval. In *Adaptive Multimedia Retrieval*, pages 12–23. 112

Schneider, J. M., Salazar, M. G., Martínez, P., and Fernández, J. L. M. (2011). Some experiments in evaluating asr systems applied to multimedia retrieval. In Detyniecki, M., García-Serrano, A., and Nrnberger, A., editors, *Adaptive Multimedia Retrieval. Understanding Media and Adapting to the User*, volume 6535 of *Lecture Notes in Computer Science*, pages 12–23. Springer Berlin Heidelberg. 214

Segura-Bedmar, I., Peña González, S., and Martínez, P. (2014a). Extracting drug indications and adverse drug reactions from spanish health social media. In *Proceedings of BioNLP 2014*, BioNLP 2014, pages 98–106. 183, 191, 205

Segura-Bedmar, I., Revert, R., and Martínez, P. (2014b). Detecting drugs and adverse events from spanish social media streams. In *Proceedings 5th International Workshop on Health Document Text Mining and Information Analysis*, EACL 2014. 183, 189, 190, 191, 196, 204, 208

Shah, U., Finin, T., Joshi, A., Cost, R. S., and Matfield, J. (2002). Information retrieval on the semantic web. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, pages 461–468, New York, NY, USA. ACM. 52, 82

Shaw, J. A. and Fox, E. A. (1994). Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2*, pages 243–252. 53

Shen, X., Tan, B., and Zhai, C. (2005). Ucair: A personalized search toolbar. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 681–681, New York, NY, USA. ACM. 56, 62, 68

Shen, X. and Zhai, C. X. (2003). Exploiting query history for document ranking in interactive information retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 377–378, New York, NY, USA. ACM. 23, 27, 56

Shokouhi, M. and Zobel, J. (2009). Robust result merging using sample-based score estimates. *ACM Trans. Inf. Syst.*, 27(3):14:1–14:29. 48

Si, L. and Callan, J. (2003). A semisupervised learning method to merge search engine results. *ACM Trans. Inf. Syst.*, 21(4):457–491. 47

Silberschatz, A., Galvin, P. B., and Gagne, G. (2008). *Operating System Concepts*. Wiley Publishing, 8th edition. 50, 128

Singh, R., Seltzer, M. L., Raj, B., and Stern, R. M. (2001). Speech in noisy environments: robust automatic segmentation, feature extraction, and hypothesis combination. In *ICASSP*, pages 273–276. IEEE. 39, 43

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380. 81

Smiley, D. and Pugh, E. (2009). *Solr 1.4 Enterprise Search Server*. Packt Publishing. 105

Smith, C. and Stavri, P. (2005). Consumer Health Vocabulary. pages 122–128. 208

Smith, R. (2007). An overview of the tesseract ocr engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02*, ICDAR '07, pages 629–633, Washington, DC, USA. IEEE Computer Society. 110

Srihari, R., Rao, A., Han, B., Munirathnam, S., and Wu, X. (2000). A model for multimodal information retrieval. In *IEEE International Conference on Multimedia and Expo (II) 2000*, pages 701–704. IEEE. v, 24, 35, 36

Steichen, B., Carenini, G., and Conati, C. (2013). User-adaptive information visualization: Using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 317–328, New York, NY, USA. ACM. 60, 62

Steiner, T., Sutton, L., Spiller, S., Lazzaro, M., Nucci, F., Croce, V., Massari, A., Camurri, A., Verroust-Blondet, A., Joyeux, L., Etzold, J., Grimm, P., Mademlis, A., Malassiotis, S., Daras, P., Axenopoulos, A., and Tzovaras, D. (2012). I-search: a multimodal search engine based on rich unified content description (rucod). In Mille, A., Gandon, F. L., Misselis, J., Rabinovich, M., and Staab, S., editors, *WWW (Companion Volume)*, pages 291–294. ACM. 30, 31, 48, 67

Strang, G. (2006). *Linear Algebra and Its Applications*. Thomson Brooks/Cole. 87

Suditu, N. and Fleuret, F. (2011). Heat: Iterative relevance feedback with one million images. Idiap-RR Idiap-RR-33-2011, Idiap. 33, 38

Sushmita, S. (2012). Study of result presentation and interaction for aggregated search. *SIGIR Forum*, 46(1):86–87. 31, 32, 39, 42, 61, 66, 84

Tata, S. and Patel, J. M. (2007). Estimating the selectivity of tf-idf based cosine similarity predicates. *SIGMOD Rec.*, 36(2):7–12. 45

Torres, J. M. (2005). *Visual Information Retrieval through Interactive Multimedia Queries*. PhD thesis, Lancaster University. 40, 43

Vallet, D., Cantador, I., and Jose, J. (2012). Exploiting semantics on external resources to gather visual examples for video retrieval. *International Journal of Multimedia Information Retrieval*, pages 1–14. 32, 39, 43

van Der Hooft, C., Sturkenboom, M., van Grootheest, K., Kingma, H., and Stricker, B. (2006). Adverse drug reaction-related hospitalisations. *Drug Saf*, 2(29):161–168. 184

Vicente-Díez, M. T., De Pablo-Sánchez, C., Martínez, P., Moreno-Schneider, J., and Salazar, M. G. (2009). Are passages enough? the miracle team participation in qa-clef2009. In *Proceedings of the 10th Cross-language Evaluation Forum Conference on Multilingual Information Access Evaluation: Text Retrieval Experiments*, CLEF'09, pages 281–288, Berlin, Heidelberg. Springer-Verlag. 140, 215

Vicente-Díez, M. T., Moreno-Schneider, J., and Martínez, P. (2010a). Temporal information needs in respubliqa: an attempt to improve accuracy. the uc3m participation at clef 2010. In Braschler, M., Harman, D., and Pianta, E., editors, *CLEF (Notebook Papers/LABs/Workshops)*. 215

Vicente-Díez, M. T., Moreno-Schneider, J., and Martínez, P. (2010b). Uc3m system: Determining the extent, type and value of time expressions in tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 329–332, Uppsala, Sweden. Association for Computational Linguistics. 215

Wang, P. and Smeaton, A. (2012). Semantics-based selection of everyday concepts in visual lifelogging. *International Journal of Multimedia Information Retrieval*, 1(2):87–101. 24, 40

Wester, K., Jnsson, A., Spigset, O., Druid, H., and Staffan, H. (2008). Incidence of fatal adverse drug reactions: a population based study. *Brit J Clin Pharmaco*, 4(65):573–579. 184

Womser-Hacker, C. (1996). *Das MIMOR-Modell: Mehrfachindexierung zur dynamischen Methoden-Objekt-Relationierung im Information Retrieval*. 58, 62, 68, 163, 178, 211

Wong, K.-M., Cheung, K.-W., and Po, L.-M. (2005). Mirror: an interactive content based image retrieval system. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 1541 – 1544 Vol. 2. 33, 38

Worring, M., Snoek, C. G. M., de Rooij, O., Nguyen, G., and Smeulders, A. W. M. (2007). The mediamill semantic video search engine. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1213–IV–1216. 52, 82, 85

Wu, S. and Crestani, F. (2015). A geometric framework for data fusion in information retrieval. *Information Systems*, 50(0):20 – 35. 50

Yang, J., Li, Q., and Zhuang, Y. (2002). Octopus: Aggressive search of multi-modality data using multifaceted knowledge base. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, pages 54–64, New York, NY, USA. ACM. 9, 30, 31, 36, 41, 49, 50, 59, 67, 68, 211

Yang, L., Cai, Y., Hanjalic, A., Hua, X.-S., and Li, S. (2012). Searching for images by video. *International Journal of Multimedia Information Retrieval*, pages 1–13. 33

Yilmaz, T., Gulen, E., Yazici, A., and Kitsuregawa, M. (2012). A relief-based modality weighting approach for multimodal information retrieval. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, ICMR '12, pages 54:1–54:8, New York, NY, USA. ACM. 9, 25, 67

Zukerman, I. and Albrecht, D. W. (2001). Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11(1-2):5–18. 152